



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Escuela de Ingeniería Informática



# VIDEOJUEGO DE MEMORIA PARA REALIDAD VIRTUAL Y AUMENTADA CON LAS OCVLUS RIFT

Trabajo de fin de grado



Tutor: Agustín Trujillo Pino

Co-Tutor de empresa: Tobias Kammann

Autor: Óscar García Sierra

2016-2017

## Índice de contenido

Agradecimientos.....	1
Resumen .....	2
Abstract .....	2
1 Introducción.....	3
1.1 Estructura del documento .....	3
1.2 Objetivos .....	4
1.3 Planificación inicial.....	5
1.4 Justificación de las competencias específicas .....	10
1.5 Metodología de desarrollo .....	11
2 Realidad Virtual .....	13
2.1 ¿Qué es?.....	13
2.2 Historia .....	13
2.3 Software.....	14
2.4 Hardware .....	15
3 Análisis.....	17
3.1 Estudio de aplicaciones similares .....	17
3.2 Requisitos.....	17
3.3 Diagrama de casos de uso .....	19
3.4 Mockups.....	22
3.5 Modelo de negocio .....	29
3.6 Normativa y legislación.....	30
3.6.1 Reglamento que afecta al TFG .....	30
3.6.2 Licencias del software .....	30
4 Desarrollo .....	33
4.1 Alcance de la implementación.....	33
4.2 Diseño arquitectónico.....	37
4.3 Modelo de la base de datos.....	38
4.4 Tecnologías .....	41
4.4.1 C#.....	42

4.4.2 Unity Personal .....	42
4.4.3 SketchUp .....	43
4.4.4 Visual Studio .....	44
4.5 Pruebas .....	45
4.6 Despliegue.....	46
4.6.1 Despliegue para equipos de sobremesa .....	47
4.6.2 Despliegue para equipos móviles .....	48
5 Conclusiones .....	51
5.1 Líneas futuras.....	52
6 Anexos .....	53
6.1 Anexo I: Manual de usuario .....	53
6.2 Anexo II: Referencias .....	58
6.2.1 Consultas terminológicas .....	58
6.2.2 Recursos: Software.....	58
6.2.3 Hardware.....	58
6.2.4 Tutoriales.....	59
6.2.5 Normativa y legislación .....	59
6.3 Anexo III: Logos .....	59

## Índice de ilustraciones

Ilustración 1: Planificación inicial detallada. ....	7
Ilustración 2: Diagrama de la planificación inicial del proyecto.....	8
Ilustración 3: Gafas de VR Oculus Rift .....	13
Ilustración 4 : Gafas de VR PlayStation.....	13
Ilustración 5: Componentes de las Oculus Rift, cámaras, mandos y gafas. ....	15
Ilustración 6 : Gafas de VR de Samsung y Oculus, Gear VR.....	16
Ilustración 7: Diagrama de casos de uso .....	19
Ilustración 8: Primera versión del proyecto .....	22
Ilustración 9: Menú del prototipo .....	23
Ilustración 10: Configuración de la partida impresa por consola de Unity.....	23
Ilustración 11: Boceto a mano de la vista superior de la escena. ....	24
Ilustración 12 : Modelo 3D del escenario, visto desde arriba. ....	25
Ilustración 13: Modelo 3D del escenario, visto en perspectiva. ....	25
Ilustración 14: Modelo 3D de la mesa de juego. ....	26
Ilustración 15: Boceto a mano de la tabla de puntuaciones. ....	26
Ilustración 16: Modelo 3D de la tabla de puntuaciones. ....	27
Ilustración 17: Modelo 3D señal de tutoriales. ....	28
Ilustración 18: Transportador enfocado con la mirada.....	33
Ilustración 19: Panel de control, habitación número 1.....	34
Ilustración 20: Consola para abrir el menú de puntuaciones. ....	35
Ilustración 21: Panel de máximas puntuaciones. ....	35
Ilustración 22: Panel de tutorial. ....	36
Ilustración 23: Menú interno del juego.....	36
Ilustración 24: Diagrama de clases de la estructura de datos.....	39
Ilustración 25: Estructura de un archivo XML .....	40
Ilustración 26: Archivo SaveGame.bin abierto desde el Notepad++. ....	41
Ilustración 27: Interfaz gráfica predeterminada de Unity.....	43
Ilustración 28: Interfaz de usuario del SketchUp. ....	44
Ilustración 29: Interfaz de usuario de Visual Studio predeterminada. ....	45
Ilustración 30: Panel de despliegue de Unity. ....	47
Ilustración 31: Dialogo de External Tools .....	49
Ilustración 32: Página web de Oculus para firmar las aplicaciones. ....	49
Ilustración 33: Portada del manual de usuario. ....	54
Ilustración 34: Primera página del manual, con la instalación y menú principal.....	55

Ilustración 35: Segunda página del manual, con el menú de partida. ....	56
Ilustración 36: Tercera página del manual, con los controles. ....	57
Ilustración 37: Primera versión del logo. ....	60
Ilustración 38: Segunda versión del logo. ....	61
Ilustración 39: Icono de la segunda versión del logo. ....	61

## Índice de tablas

Tabla 1: Planificación inicial.....	6
Tabla 2: Estado de la planificación final. ....	9
Tabla 3: Comparativa de otras aplicaciones en el mercado.....	17
Tabla 4: Tabla de especificación de Abrir menú.....	20
Tabla 5: Tabla de especificación de Completar nivel. ....	21
Tabla 6: Tabla de especificación de Gestionar partida.....	21
Tabla 7: Lean Canvas .....	29



## Agradecimientos

A mi tutor Agustín Trujillo Pino por aconsejarme y ayudarme durante todo el transcurso del Trabajo de Fin de Grado.

A todos mis compañeros que me han acompañado durante estos 4 años de grado, pues sin ellos nunca hubiera llegado hasta aquí.

A mi padre, mi hermana y mi pareja, por siempre estar cuando más se les necesita y sin fallarme en ningún momento.

A la empresa de edataconsulting por aportar el material necesario, a todos los miembros que la forman, por tratarme como a uno más y en especial a Tobias Kammann por su implicación y su gran atención en todo momento, sin el cual todo esto no hubiera sido posible.

A Guillermo Cubero, Carla Cabestrero, Adrián Louro y Saskia Santana, por aportar ideas al proyecto, probar las demos del juego, su ánimo y su apoyo en cada momento.

## Resumen

El proyecto se ha realizado en colaboración con la empresa edataconsulting<sup>1</sup> y consiste en crear un videojuego en Unity, para Realidad Virtual, en un escenario 3D. Con este proyecto, se busca aprender a utilizar las nuevas formas de interacción humano-máquina y aprender a utilizar el motor gráfico de Unity. Las principales características son, una tabla de mejores puntuaciones, guardado de partida automático, diferentes niveles de dificultad, posibilidad de añadir diferentes tipos de juego con facilidad.

## Abstract

The project has been done in collaboration with the edataconsulting company and consists in create a video game in Unity, for Virtual Reality, in a 3D scenery. With this project, you will learn to use the new forms of human-machine interaction and learn how to use the Unity graphics engine. The main features are, a scoreboard of best scores, automatic save of the game, different levels, easy way to add new types of games.

---

<sup>1</sup> Empresa colaboradora, fue la responsable de ceder las herramientas necesarias para el desarrollo del proyecto. Fundada en 2003 y con sede en Gelsenkirchen (Alemania).

## 1 Introducción

### 1.1 Estructura del documento

En el presente documento se ha intentado seguir un orden lógico, que haga fácil su lectura y entendimiento por parte del lector.

A lo largo de estas líneas se pueden introducir términos que, en el caso de ser técnicos, intentan ser explicados para poder llegar al mayor público posible. También por ello se han incluido abundantes ilustraciones, para permitir una mayor asimilación de los textos que acompañan.

A continuación, se encuentran resumimos los principales temas abordados y que encontrará el lector durante su lectura:

- **Introducción:**

Es el apartado donde nos encontramos ahora. En este apartado queremos poner en situación al lector, para ello se pasa a explicar lo principales objetivos del proyecto, como se planificó en un principio y las metodologías utilizadas durante el desarrollo de este mismo. Sin este apartado sería complicado que el lector supiera situarse.

- **Realidad Virtual:**

En este apartado pasaremos a explicar brevemente en que consiste la tecnología de Realidad Virtual, de que se compone, que es necesario para desarrollar para esta tecnología, como ha ido cambiando y que diferentes tipos de dispositivos existen. Todo ello para que cualquier lector pueda continuar sin ningún tipo de problema.

- **Análisis:**

Con el análisis pretendemos estudiar la viabilidad del proyecto, las diferentes opciones existentes y la mejor manera de llevar adelante nuestro trabajo. También se mostrarán los prototipos y diferentes diseños que se realizaron, así como el modelo de negocio, las normativa y legislaciones a cumplir.

- **Desarrollo:**

Aquí se pasa a explicar cómo fue la fase de implementación del proyecto, en ella daremos paso al lector a ver hasta donde llegó el proyecto. Para ello será necesario mostrar cómo se ha estructurado, que tecnologías se utilizaron, pruebas realizadas y como se debe desplegar.
- **Conclusiones:**

Después de haber comentado todo el camino recorrido hasta llegar aquí, estaremos en disposición de explicar mejoras y propuestas, que puedan engrandecer nuestro trabajo. Además, se realizará una reflexión sobre cómo ha sido esta experiencia.
- **Documentos:**

Se pondrá a disposición del lector las fuentes consultadas durante la elaboración de este proyecto, por si necesita obtener más información sobre algún tema en concreto.
- **Anexos:**

Por último, pero no menos importante, los contenidos que por su extensión no se han introducido en mitad del documento, para facilitar la lectura y reducir la posible complejidad del documento, pero no por ello son menos relevante.

## 1.2 Objetivos

Los objetivos propuestos a alcanzar durante el desarrollo del proyecto son los que se exponen a continuación:

- **Estudio y aprendizaje de la herramienta de desarrollo de videojuegos Unity.**

Como ya se dijo con anterioridad, para poder desarrollar en VR es necesario un motor gráfico 3D. Será necesario aprender a utilizarlo, gracias a los tutoriales proporcionados por Unity, este objetivo fue fácil de llevar. También poseen un foro, donde es fácil de resolver dudas y encontrar soluciones. La mayor complicación era encontrar la solución para la versión correcta de Unity.

- **Estudio y aprendizaje de la tecnología de Realidad Virtual.**  
Durante los primeros días, estuve probando las diferentes aplicaciones disponibles para los diferentes dispositivos, con la finalidad de ver que posibilidades y que limitaciones tenía esta tecnología.
- **Análisis y diseño de las mecánicas del juego.**  
La idea principal del juego está enfocada en ejercitar la memoria del jugador. Se le aplico un concepto más competitivo, al añadirle:
  - Cuenta atrás, una vez finalizado ese tiempo, la partida termina.
  - Diferentes dificultades, más objetos que memorizar.
  - Puntuación, cuanto antes lo termines y más combos<sup>2</sup> consigas mejor.
- **Desarrollo de la arquitectura del programa.**  
Unity está enfocado con una arquitectura por componentes, y cada objeto es independiente al resto. Este tipo de arquitectura permite crear componentes y reutilizarlos en diferentes partes del desarrollo o incluso en otros desarrollos. Unity permite importar y exportar componentes en forma de asset<sup>3</sup>.
- **Implementación del videojuego siguiendo los estudios y análisis previos.**  
Para el desarrollo de la aplicación se fueron siguiendo los estudios realizados. Cuando se conseguía una versión funcional, se ponía a prueba para encontrar posibles fallos de la implementación y se mostraba a diferentes usuarios para obtener diferentes opiniones.

### 1.3 Planificación inicial

En **Tabla 1** se definen, inicialmente, las horas estimadas que fueron aplicadas a cada fase del proyecto:

---

<sup>2</sup> Al realizar aciertos sucesivos se aumenta un multiplicador de puntuación, una vez se falla, este multiplicador vuelve a 1.

<sup>3</sup> Paquete que se puede exportar desde Unity e importarlo en otros proyectos. Este paquete es totalmente independiente y cuando se exporta, recoge todas las dependencias para poder funcionar por si solo en otro proyecto. Estos también se pueden obtener a partir de una tienda, Asset Store, dentro de Unity.

Fases	Duración Estimada	Tareas
<b>Estudio previo / Análisis</b>	60	<b>Tarea 1.1:</b> Estudio del funcionamiento de Unity 5.5
		<b>Tarea 1.2:</b> Estudio del funcionamiento del resto de aplicaciones realizadas en Realidad Virtual
		<b>Tarea 1.3:</b> Estudio de las posibilidades que permite la Realidad Virtual
<b>Diseño / Desarrollo / Implementación</b>	130	<b>Tarea 2.1:</b> Diseño de las mecánicas del juego.
		<b>Tarea 2.2:</b> Diseño de la arquitectura del programa
		<b>Tarea 2.3:</b> Diseño del escenario 3D, sus animaciones y sonidos.
		<b>Tarea 2.4:</b> Implementación de las mecánicas.
<b>Evaluación / Validación / Prueba</b>	70	<b>Tarea 3.1:</b> Comprobación de las mecánicas del juego.
		<b>Tarea 3.2:</b> Comprobación de las animaciones y sonidos.
		<b>Tarea 3.3:</b> Comprobación de integridad y solidez.
<b>Documentación / Presentación</b>	40	<b>Tarea 4.1:</b> Realización de la memoria
		<b>Tarea 4.2:</b> Realización de la presentación.

*Tabla 1: Planificación inicial*

Una vez comenzado el proyecto se me recomendó que realizara una planificación más detallada en ProjectLibre<sup>4</sup> de lo que sería el proyecto, para así poder realizar una serie de reuniones a lo largo del proyecto y poder comprobar que todo continuaba según lo previsto. La primera versión es la que se puede observar en la **Ilustración 1 e Ilustración 2**.

	📌	Nombre	Duración	Inicio	Terminado	Predecesores
1	📌	Setup PC	1 day	23/01/17 8:00	23/01/17 17:00	
2		Setup VR + Trials	2 days	24/01/17 8:00	25/01/17 17:00	1
3		Concept	2 days	26/01/17 8:00	27/01/17 17:00	2
4	☐	<b>Unity Basic Level Design</b>	<b>4 days?</b>	<b>30/01/17 8:00</b>	<b>2/02/17 17:00</b>	<b>3</b>
5		Single Play Desing	1 day?	30/01/17 8:00	30/01/17 17:00	
6		SandBox Desing	1 day?	31/01/17 8:00	31/01/17 17:00	5
7		Multiple Rooms Desing	2 days?	1/02/17 8:00	2/02/17 17:00	6
8		Learn VR Interaction on tests	3 days	3/02/17 8:00	7/02/17 17:00	4
9	☐	<b>Basic Game Logic (Select, Uncover Boxes)</b>	<b>15 days?</b>	<b>8/02/17 8:00</b>	<b>28/02/17 17:00</b>	<b>8</b>
10		MenuController	2 days?	8/02/17 8:00	9/02/17 17:00	
11		Single Game Controller	2 days?	10/02/17 8:00	13/02/17 17:00	10
12	☐	<b>Multiple rooms Controller</b>	<b>11 days?</b>	<b>14/02/17 8:00</b>	<b>28/02/17 17:00</b>	<b>11</b>
13		Save Game And Scores	1 day?	14/02/17 8:00	14/02/17 17:00	
14		Logic First Room	5 days	15/02/17 8:00	21/02/17 17:00	13
15		Logic Second Room	5 days	22/02/17 8:00	28/02/17 17:00	14
16		Testing & Bugfixing	3 days	1/03/17 8:00	3/03/17 17:00	9
17	☐	<b>Initial visual enhancements</b>	<b>2 days?</b>	<b>6/03/17 8:00</b>	<b>7/03/17 17:00</b>	<b>16</b>
18		Redesing Single Game	1 day?	6/03/17 8:00	6/03/17 17:00	
19		Decorate Second Room	2 days	6/03/17 8:00	7/03/17 17:00	
20		Decorate First Room	2 days?	6/03/17 8:00	7/03/17 17:00	
21	☐	<b>Advanced Game Logic</b>	<b>5 days</b>	<b>8/03/17 8:00</b>	<b>14/03/17 17:00</b>	<b>17</b>
22		SandBox Logic	5 days	8/03/17 8:00	14/03/17 17:00	
23	☐	<b>Advanced Visual Enhancements</b>	<b>5 days?</b>	<b>15/03/17 8:00</b>	<b>21/03/17 17:00</b>	<b>21</b>
24		Animation	2 days?	15/03/17 8:00	16/03/17 17:00	
25		Particles	1 day?	17/03/17 8:00	17/03/17 17:00	24
26		Sound	1 day?	20/03/17 8:00	20/03/17 17:00	25
27		Light	1 day?	21/03/17 8:00	21/03/17 17:00	26
28		Additional Testing & Bugfixing	5 days	22/03/17 8:00	28/03/17 17:00	23
29		Documentation	10 days	29/03/17 8:00	11/04/17 17:00	28
30		Presentation	1 day?	12/04/17 8:00	12/04/17 17:00	29
31		Milestone - VR Interaction in own world done	0 days	7/02/17 17:00	7/02/17 17:00	8
32	📌	Milestone - Game Logic Basics done in VR	0 days	1/03/17 17:00	1/03/17 17:00	9
33		Milestone - Visual Enhancements done	0 days	7/03/17 17:00	7/03/17 17:00	17
34		Milestone - Advanced Game Logic Done	0 days	14/03/17 17:00	14/03/17 17:00	21
35		Milestone - All Coding done	0 days	28/03/17 17:00	28/03/17 17:00	28
36		Milestone - Documentation done	0 days	11/04/17 17:00	11/04/17 17:00	29
37		Presentation - Game Logic done	0 days	3/03/17 17:00	3/03/17 17:00	16
38		Presentation - Advanced Logic done	0 days	14/03/17 17:00	14/03/17 17:00	21
39		Presentation - Final Game Demo	0 days	12/04/17 17:00	12/04/17 17:00	30

*Ilustración 1: Planificación inicial detallada.*

<sup>4</sup> Es un software de administración de proyectos de código abierto, similar a Microsoft Project. Este corre sobre la plataforma Java, lo que permite ejecutarlo en varios sistemas operativos. Fue lanzado en agosto de 2012.

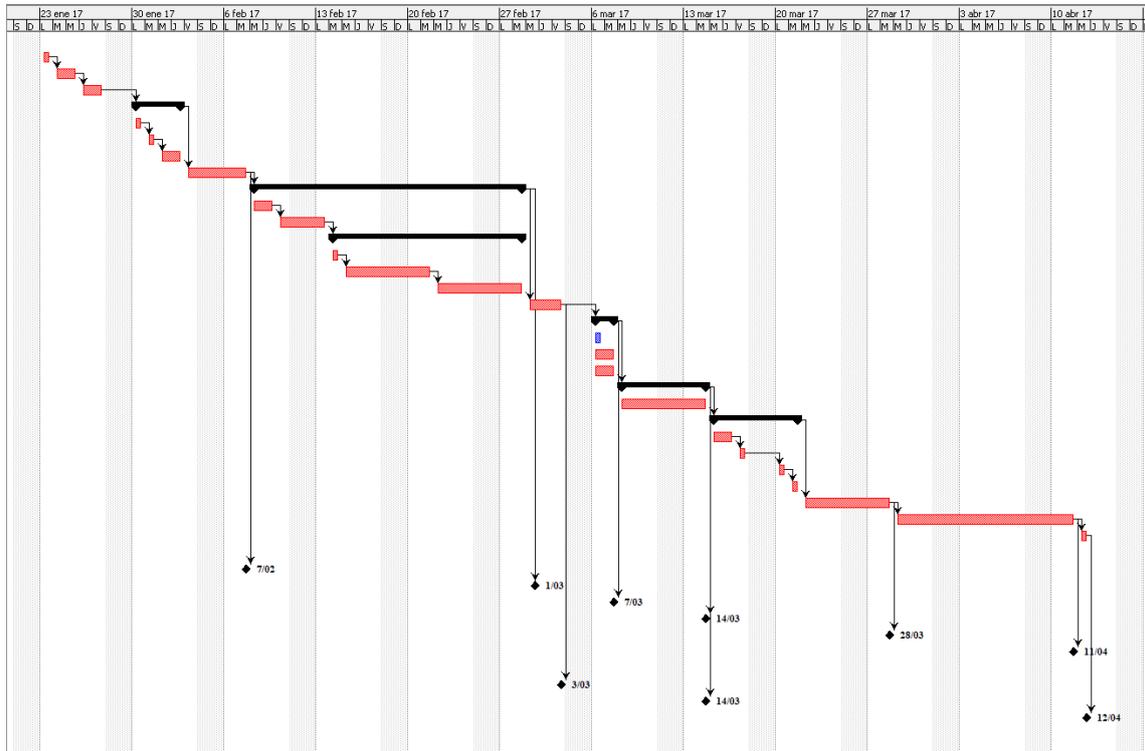


Ilustración 2: Diagrama de la planificación inicial del proyecto.

Los “milestone” son fechas en las que se realizaban reuniones, para comprobar que el proyecto seguía avanzando de forma correcta. Se revisaba lo que se había desarrollado durante ese tiempo, corrigiendo problemas, añadían cambios y mejoras al proyecto.

En la **Tabla 2** se pasa a mostrar una planificación más exacta del proyecto, luego se explicará, porque surgen estas diferencias con respecto a la primera estimación.

Fases	Duración Estimada	Tareas
Estudio previo / Análisis	80	<b>Tarea 1.1:</b> Realización de los tutoriales de Unity 5.4.
		<b>Tarea 1.2:</b> Instalación del equipo y de las aplicaciones necesarias para el desarrollo del proyecto.
		<b>Tarea 1.3:</b> Estudio del funcionamiento del resto de aplicaciones realizadas en Realidad Virtual

		<b>Tarea 1.4:</b> Estudio de las posibilidades que permite la Realidad Virtual
<b>Diseño / Desarrollo / Implementación</b>	200	<b>Tarea 2.1:</b> Diseño de las mecánicas del juego.
		<b>Tarea 2.2:</b> Diseño de la arquitectura del programa
		<b>Tarea 2.3:</b> Diseño del escenario 3D, sus animaciones y sonidos.
		<b>Tarea 2.4:</b> Implementación de las mecánicas.
		<b>Tarea 2.5:</b> Mejoras visuales, sonido, lógica.
<b>Evaluación / Validación / Prueba</b>	80	<b>Tarea 3.1:</b> Comprobación de las mecánicas del juego.
		<b>Tarea 3.2:</b> Comprobación de las animaciones y sonidos.
		<b>Tarea 3.3:</b> Comprobación de integridad y solidez.
<b>Documentación / Presentación</b>	60	<b>Tarea 4.1:</b> Realización de la memoria
		<b>Tarea 4.2:</b> Realización de la presentación.

*Tabla 2: Estado de la planificación final.*

El trabajo comenzó a principios de enero de 2017, por estas fechas se comenzó a realizar los primeros tutoriales propuestos por Unity. En estos tutoriales, se explica cómo utilizar la herramienta de Unity de forma básica y como programar algo sencillito dentro del entorno. Los tutoriales se desarrollan tanto para C#, como para JavaScript.

Una vez concluido los tutoriales, se realizó la incorporación dentro de edataconsulting. Una vez allí se comenzó con la instalación de todas las herramientas necesarias para poder continuar el proyecto. A parte, se pudo probar las diferentes aplicaciones que se encontraban en aquel momento en el mercado para poder conocer mejor las posibilidades que ofrecía. Estas tareas, llevaron un poco más tiempo del esperado.

En la fase de diseño, no hubo una gran dificultad a la hora de ajustarse a lo previsto en un principio, pero si que se produjeron diferentes adversidades durante el desarrollo que produjeron que se necesitara más tiempo para poder concluir esta fase,

pues el lenguaje y la tecnología utilizada eran totalmente desconocidas. Una vez resueltas estas adversidades se pudo continuar con el correcto desarrollo de esta misma.

En las dos últimas fases se necesitó un poco más de tiempo, para así poder asegurar un trabajo sólido y fiable.

#### 1.4 Justificación de las competencias específicas

Las tareas de investigación, análisis, diseño e implementación llevadas a cabo han permitido el desarrollado de las siguientes competencias propuestas para alcanzar la asignatura de Trabajo de Fin de Grado:

- **FB01.** Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal, cálculo diferencial e integral, métodos numéricos, algorítmica numérica, estadística y optimización.
  - **Justificación FB01.**

Durante el desarrollo del proyecto se plantearon problemas espaciales, al tener que situar objetos distribuidos de forma equidistante en una semicircunferencia, los cuales, no eran siempre la misma cantidad.
- **CIIO1.** Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
  - **Justificación CIIO1.**

Como se puede ver en este documento y en el proyecto en general, el desarrollo del trabajo es propio de un ingeniero. Los pasos dados, así como las soluciones propuestas manifiestan la necesidad del uso de los conocimientos y las capacidades propios de nuestra área de estudio.
- **CIIO6.** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
  - **Justificación CIIO6.**

Durante la ejecución de cualquier videojuego, es importante no realizar procesos muy pesados, para evitar que el usuario pueda experimentar lentitud durante la ejecución de este mismo.

- **CII07.** Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
  - **Justificación CII07**  
Mirar justificación CII01
  
- **CII16.** Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
  - **Justificación CII16.**  
Mirar justificación CII01
  
- **CII17.** Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.
  - **Justificación CII17.**  
Nos encontramos ante un proyecto en Realidad Virtual, por lo tanto, las interfaces de usuario deben estar adaptadas a este tipo de interacción humano-maquina.
  
- **IS04.** Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.
  - **Justificación IS04.**  
Mirar justificación CII01.

### 1.5 Metodología de desarrollo

Para la realización de este trabajo se ha seguido el modelo de ciclo de vida en espiral, el cual consiste en dividir el desarrollo en ciclos o iteraciones y cada ciclo en cuatro fases o tareas. El número de iteraciones de esta metodología variará según los requisitos definidos y finalizará cuando se logre un producto mínimo viable.

Antes de comenzar un nuevo ciclo, se comprobaban los objetivos cumplidos en las fases anteriores. Tras esto, se establecerá el punto de partida y la meta a conseguir en la nueva fase. Cada nuevo ciclo terminado, generará una nueva versión de la aplicación, que será más completa que la anterior. Cada fase está planificada con la siguiente estructura:

- **Determinar los objetivos.**

En esta tarea se define la meta a lograr en esta iteración. Así se identifican las limitaciones del ciclo y sus riesgos.

- **Análisis del riesgo.**

Se analizan los riesgos que se presentan en la fase y se definen nuevas estrategias que permitan minimizar los riesgos encontrados.

- **Desarrollar, verificar y validar.**

El desarrollo se divide en tareas. Esto permite cubrir todos los objetivos fijados. A continuación, se comienza con el desarrollo, realizando las tareas planificadas y, finalmente, probar y verificar los objetivos.

- **Planificar.**

Finalmente se revisa el proyecto y se decide si continuar con la iteración actual. Se comprobará los objetivos marcados para ese ciclo, se revisará los requisitos y, si fuera necesario, se mejoraría la funcionalidad implementada. Tras superar este punto, se comenzará una nueva fase.

Tras múltiples iteraciones, se habrá obtenido un producto mínimo viable que cumplirá con los requisitos definidos en un principio.

## 2 Realidad Virtual

### 2.1 ¿Qué es?

La Realidad Virtual o VR de sus siglas del inglés Virtual Reality, son dispositivos que te permiten introducirte dentro de un entorno totalmente creado por ordenador y que la forma de interactuar con la maquina sea lo más natural posible, como por ejemplo cuando quieres mirar para un lado, no uses un ratón, si no que gires la cabeza. Estos dispositivos, pueden ir de los más sencillo, que sería solamente unas gafas, a incluso dispositivos que te permiten capturar tu movimiento dentro de una sala y el movimiento de tus manos, para ello se necesitan cámaras y unos mandos especiales, que permitan realizar un seguimiento preciso de los movimientos realizados.

### 2.2 Historia

A partir de 2016, la Realidad Virtual ha sufrido un crecimiento muy espontaneo, con la aparición de los primeros dispositivos para el consumidor, Oculus Rift (**Ilustración 3**), seguido de las HTC Vive, después de esos, empezaron a aparecer más gafas de otras compañías como las de Microsoft, las HoloLens, pero estas son de Realidad Aumentada y Sony con sus PlayStation VR (**Ilustración 4**), desarrolladas en exclusiva para su última consola, la PlayStation 4 Pro.



*Ilustración 3: Gafas de VR Oculues Rift*



*Ilustración 4 : Gafas de VR PlayStation*

Con la aparición de todos estos dispositivos y otros menos conocidos, la tecnología ha ido evolucionando de forma muy rápida y ha ido introduciéndose en el

mercado. Donde mayor acogida ha tenido, es en el sector de los videojuegos, pero es cierto que también tiene un gran impacto en el sector empresarial, pudiendo realizar algunas tareas de una forma más sencilla y eficiente, ahorrando así costes y tiempo.

### 2.3 Software

Para poder comenzar a desarrollar, cualquier tipo de aplicación, para VR es necesario tener una serie de programas mínimos:

- Motor gráfico 3D. En nuestro caso, se hizo uso de Unity 5.5 y 5.6, pues es gratuito y está muy bien documentado, aunque ya hablaré más adelante de Unity. Es imprescindible tener un motor gráfico<sup>5</sup>, debido a que debemos de generar un escenario donde nuestro usuario se encontrará y también será donde se ejecutó toda nuestra aplicación.
- Programa de diseño 3D. En este ámbito existen diferentes alternativas, desde aplicaciones muy potentes y profesionales como Maya, que es de pago, otra como Blender, que es gratuita, pero no opté por ninguna de estas dos, pues no tenía experiencia en ninguna de ellas, así que opté por SketchUp, en el cual ya tenía algo de experiencia y es mucho más sencillo de utilizar, aunque no tiene el mismo número de herramientas que cualquiera de las otras nombradas.
- Entorno de desarrollo. Unity, permite elegir entre dos lenguajes a la hora de programar su lógica. Podemos optar por JavaScript o C#. En nuestro caso optamos por C#, por ser el más parecido a Java (lenguaje que se imparte durante el grado) y, por lo tanto, se debe desarrollar en el entorno propio de Microsoft, el Visual Studio.

---

<sup>5</sup> Es una herramienta de desarrollo, que provee al videojuego de un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico.

## 2.4 Hardware

Para poder testar bien la aplicación que se va a desarrollar y hacerse una idea de cómo va a ser la experiencia por parte del usuario es necesario tener en posesión, al menos unas gafas de VR. Como ya expliqué anteriormente, hay diferentes tipos de gafas, no hemos decantado por la opción de Oculus, pues tienes dos variantes, una para dispositivos de escritorio y otra para dispositivos móviles Samsung.

Por supuesto, las dos variantes de Oculus no son iguales y pasaremos a describirlas en las siguientes líneas:

- Dispositivo de escritorio, compuesto no solamente por las gafas, también tiene cuatro cámaras y dos mandos (**Ilustración 5**), uno para cada mano, con todo esto el dispositivo de escritorio, puede manejar diferentes formas de interactuar con la máquina, se puede tocar con una mano un botón, agarrar un objeto e incluso lanzarlo. A parte, tiene más potencia que el dispositivo móvil, por lo cual puede ejecutar juegos con mejor calidad gráfica y mayor complejidad computacional.



*Ilustración 5: Componentes de las Oculus Rift, cámaras, mandos y gafas.*

- Dispositivo móvil, compuesto solamente por unas gafas, con una pequeña zona táctil y cuatro botones, como se puede observar en la **Ilustración 6**. Es mucho más portable, pues solo es necesario llevar el móvil y las gafas donde se coloca el mismo, pero se depende de la autonomía del dispositivo. Se le puede agregar un mando Bluetooth, con el que facilitar el control.



*Ilustración 6 : Gafas de VR de Samsung y Oculus, Gear VR*

## 3 Análisis

### 3.1 Estudio de aplicaciones similares

Actualmente en el mercado hay diferentes opciones de videojuegos en Realidad Virtual. Durante el análisis, no se ha encontrado ninguna opción parecida a la que nosotros ofrecemos, pero si es cierto que toda aplicación en este entorno es un competidor y por lo tanto nos centraremos en las que se asemejan más a la nuestra.

En **Tabla 3** se puede observar una comparativa entre las aplicaciones que más se asemejan a la nuestra y que ofrecen cada una de ellas.

Aplicación	Desarrollador	Fecha de Lanzamiento	Fortalezas	Debilidades
<b>Proton Pulse</b>	Justin Moravetz	Q1 2017	Compatible con todas las plataformas de VR actuales, puntuaciones, vidas límite.	No tiene tiempo límite cada partida, produce mareo, apartado gráfico mejorable.
<b>Esper 2</b>	Coatsink	Q1 2016	Entretenido, un tutorial muy completo e integrado en el juego.	Falta de competición, controles complicados.

*Tabla 3: Comparativa de otras aplicaciones en el mercado.*

Como ya explicamos, no se encuentra en el mercado ninguna aplicación que este pensada, para ejercitar la memoria y la vez competir con uno mismo para mejorar. Esper 2, es de rompecabezas, pero no ejercitas la memoria, solamente la agilidad mental, por otro lado, Proton Pulse, si introduce una característica más competitiva, al introducir puntuaciones y vidas limite.

### 3.2 Requisitos

A continuaciones pasamos a describir los diferentes requisitos de usuario capturados durante la etapa de análisis de nuestro proyecto. Estos se han redactado

en forma de historias de usuario, pues es la mejor forma de administrar los requisitos y adaptarnos a futuros cambios.

Al ser un videojuego, solo hemos detectado un rol posible, el jugador, y del cual tratan todas las historias de usuario.

A continuación, se pasa a describir las historias de usuario de los requisitos funcionales:

- Como jugador, quiero poder continuar mi anterior partida, para no perder el progreso conseguido.
- Como jugador, quiero poder crear una nueva partida, para poder comenzar el juego de nuevo.
- Como jugador, quiero poder moverme por el mapa, para poder avanzar por los diferentes niveles.
- Como jugador, quiero poder ver mis mejores puntuaciones, para poder mejorarlas.
- Como jugador, quiero poder cambiar la dificultad de la partida, para poder adaptarla a mi nivel.
- Como jugador, quiero que el sistema guarde la partida, para no perder el progreso obtenido durante la partida.
- Como jugador, quiero que cuando consiga un acierto se me recompense, para motivarme a continuar.
- Como jugador, quiero que cuando se termine el tiempo de la partida, el espacio de juego se limpie.
- Como jugador, quiero poder activar o desactivar el control con mando, para poder moverme libremente por el mapa.

Ahora continuaremos con las historias de usuario de los requisitos no funcionales:

- Como jugador, quiero que se puede ejecutar en dispositivos GearVR
- Como jugador, quiero que se pueda ejecutar en modo escritorio.

### 3.3 Diagrama de casos de uso

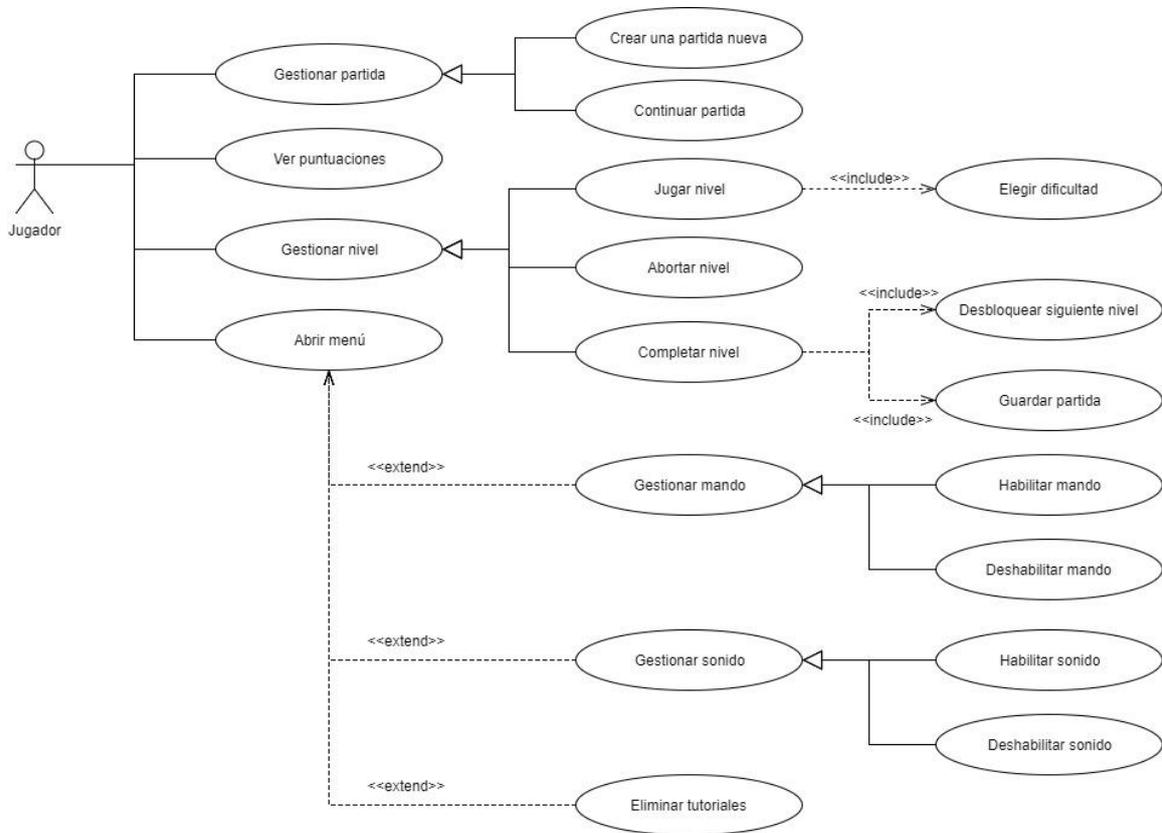


Ilustración 7: Diagrama de casos de uso

En la **Ilustración 7** se ven reflejadas las principales acciones que puede realizar el usuario, Jugador, en nuestro videojuego.

En las tablas que preceden a este párrafo (**Tabla 4**, **Tabla 5** y **Tabla 6**) pasaremos a realizar una especificación de los casos de uso más relevantes.

Caso de uso		Abrir menú	
<b>Descripción</b>	El jugador desea tener un menú donde realizar acciones.		
<b>Actor</b>	Jugador		
<b>Precondiciones</b>	El jugador debe haber iniciado una partida.		
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario debe pulsar la tecla de menú o atrás.	
<b>Postcondiciones</b>	Se despliega un menú donde podrá realizar las acciones.		
<b>Variaciones</b>	<b>Paso</b>	<b>Acción</b>	
<b>Extensiones</b>	<b>Paso</b>	<b>Acción</b>	
	1.1	El jugador puede activar o desactivar el control con mando.	
	1.2	El jugador puede salir de la partida al menú de carga.	
	1.3	El jugador puede activar o desactivar el sonido del juego.	
<b>Excepciones</b>			
<b>Observaciones</b>			

*Tabla 4: Tabla de especificación de Abrir menú.*

Caso de uso		Completar nivel	
<b>Descripción</b>	El jugador necesita poder terminar un nivel		
<b>Actor</b>	Jugador		
<b>Precondiciones</b>	El jugador debe haber terminado con éxito el nivel.		
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	Se abre la puerta del nivel terminado	
	2	Se desbloquea el siguiente nivel	
	3	Se guarda el estado del juego.	
<b>Postcondiciones</b>	Se despliega un menú donde podrá empezar de nuevo el nivel.		
<b>Variaciones</b>	<b>Paso</b>	<b>Acción</b>	
<b>Extensiones</b>	<b>Paso</b>	<b>Acción</b>	
<b>Excepciones</b>			
<b>Observaciones</b>	Para terminar con éxito un nivel se debe haber conseguido		



	terminar en el tiempo, y con una puntuación mayor a 20 puntos.
--	--

*Tabla 5: Tabla de especificación de Completar nivel.*

Caso de uso		Gestionar partida	
<b>Descripción</b>	El jugador necesita poder gestionar su partida.		
<b>Actor</b>	Jugador		
<b>Precondiciones</b>	Abrir la aplicación		
<b>Flujo normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	Se le muestra un menú donde podrá elegir su acción.	
<b>Postcondiciones</b>	Se despliega un menú donde podrá empezar de nuevo el nivel.		
<b>Variaciones</b>	<b>Paso</b>	<b>Acción</b>	
	1.1	El jugador desea crear una nueva partida.	
	1.1.1	Se busca la partida actual, si es que existe una partida y se elimina el archivo.	
	1.1.2	Se crea un nuevo archivo para guardar la partida.	
	1.1.3	Se carga el juego.	
	1.2	El jugador desea continuar la partida.	
	1.2.1	Se carga el juego.	
	1.2.2	Se restaura la partida guardada.	
<b>Extensiones</b>	<b>Paso</b>	<b>Acción</b>	
<b>Excepciones</b>			
<b>Observaciones</b>			

*Tabla 6: Tabla de especificación de Gestionar partida.*

### 3.4 Mockups

Durante el desarrollo se realizó la implementación de un prototipo de lo que sería luego la aplicación. Como podemos observar en la **Ilustración 8**. Esta versión era funcional, pero no muy atractiva a la vista, además de diversos problemas de jugabilidad, como son:

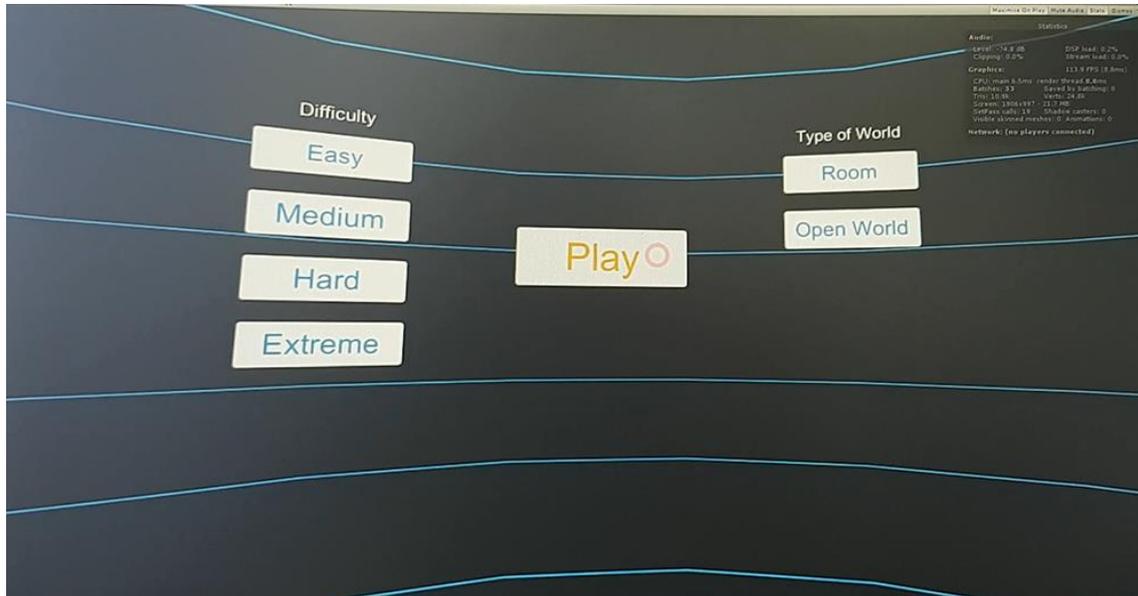
- No se le muestra en ningún momento al usuario la solución, para que la memorice.
- Al ser colores, puede producir problemas a algunos de los usuarios, pues cuando aumenta mucho el número de objetos es complicado buscar colores que se diferencien lo suficiente entre ellos.
- No se permitía el movimiento del usuario, el usuario solo podía mover la cabeza para seleccionar los objetos.
- Ya se encontraba implementado un sistema de puntuación, precario, debido a que no se almacenaba en ningún lugar las puntuaciones y solo se podía ver el resultado obtenido en cada partida por la consola de Unity.
- Como se guardaban las puntuaciones y solo estaba pensado para un tipo de juego, no se había implementado ningún tipo de gestos de partida y archivos.



*Ilustración 8: Primera versión del proyecto*

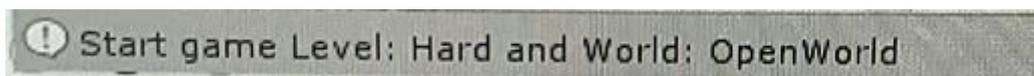
Para este prototipo se implementó un menú, que se puede observar en la **Ilustración 9**, desde donde se podía elegir el nivel de dificultad de la partida y el modo

de juego, en aquel momento solo se disponía del primero (“Room”). Este menú era totalmente funcional.



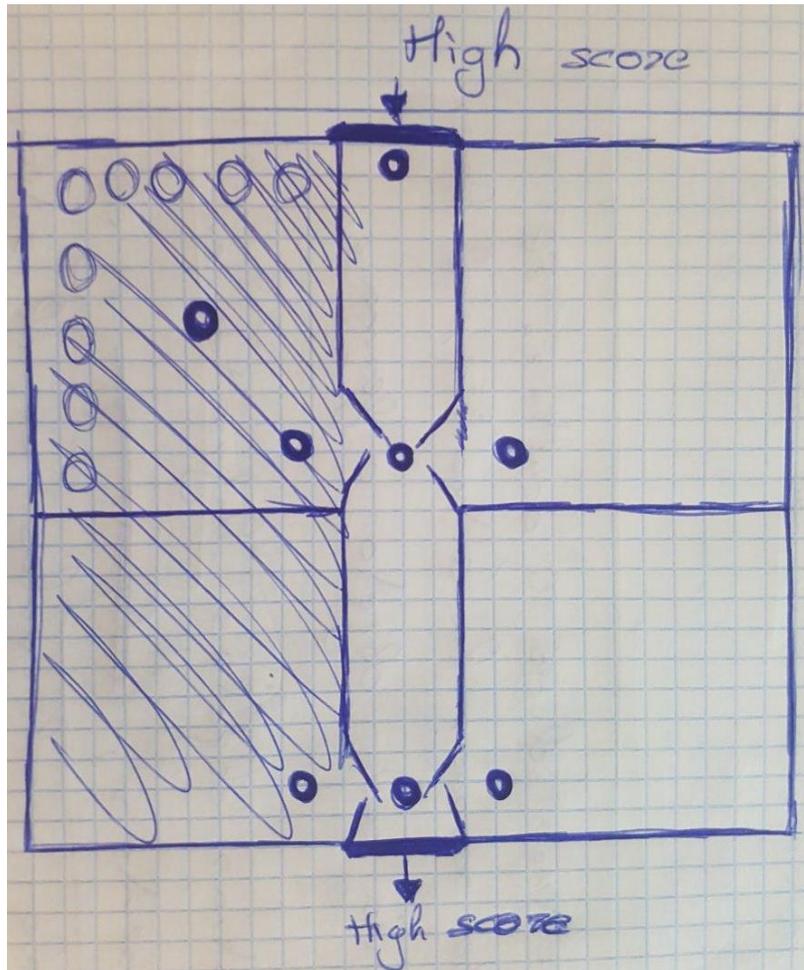
*Ilustración 9: Menú del prototipo*

Para ayudarnos a detectar problemas se imprimía por consola de Unity el tipo de dificultad y el modo de juego al pulsar el *Play*. Esto último lo podemos observar en la **Ilustración 10**. Una vez se pulsaba el botón de Play se ejecutaba la escena de la **Ilustración 8**: Primera versión del proyecto, con la configuración de nivel seleccionado.



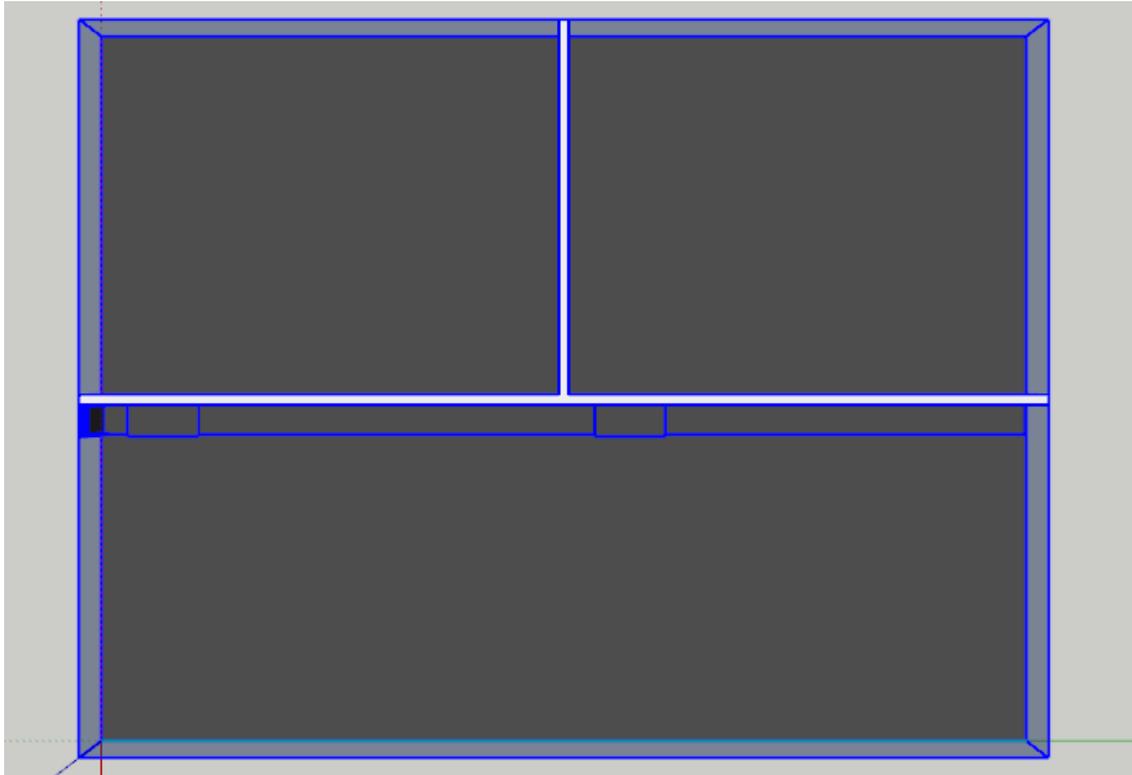
*Ilustración 10: Configuración de la partida impresa por consola de Unity*

Después de terminar este prototipo funcional, se pasó al diseño de lo que sería la zona donde el jugador aparecería y donde se colocarían los diferentes modos de juego y la tabla de puntuación. Se comenzó realizando unos bocetos a manos de lo que sería la vista aérea, como se puede ver en **Ilustración 11**.

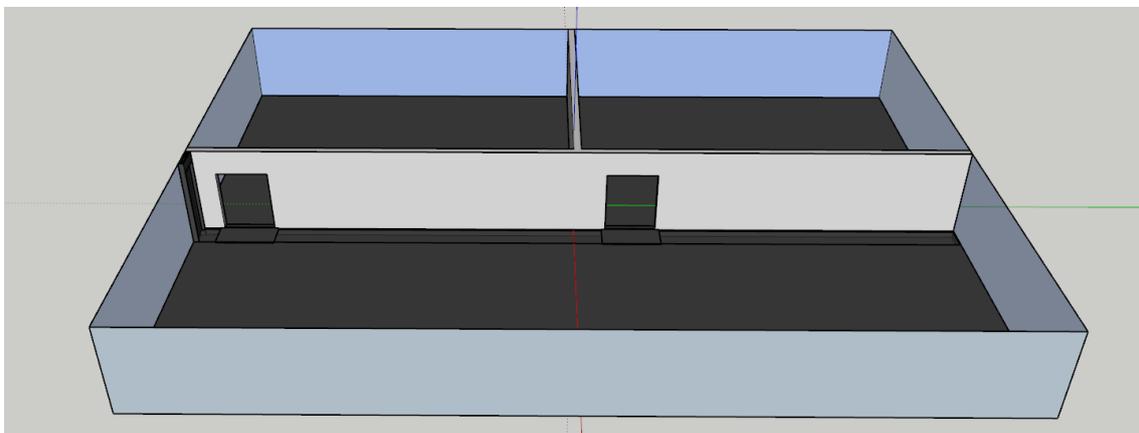


*Ilustración 11: Boceto a mano de la vista superior de la escena.*

Una vez concluido estos bocetos, se pasó a realizar un modelo 3D de lo plasmado en el boceto. Durante el modelado 3D, se eliminaron dos de las cuatro habitaciones como podemos ver en la **Ilustración 12** e **Ilustración 13**, que se pensaron en un principio, y se sustituyó por una zona por dónde desplazarse entre las habitaciones y a su vez donde poder ver la pantalla de puntuaciones.



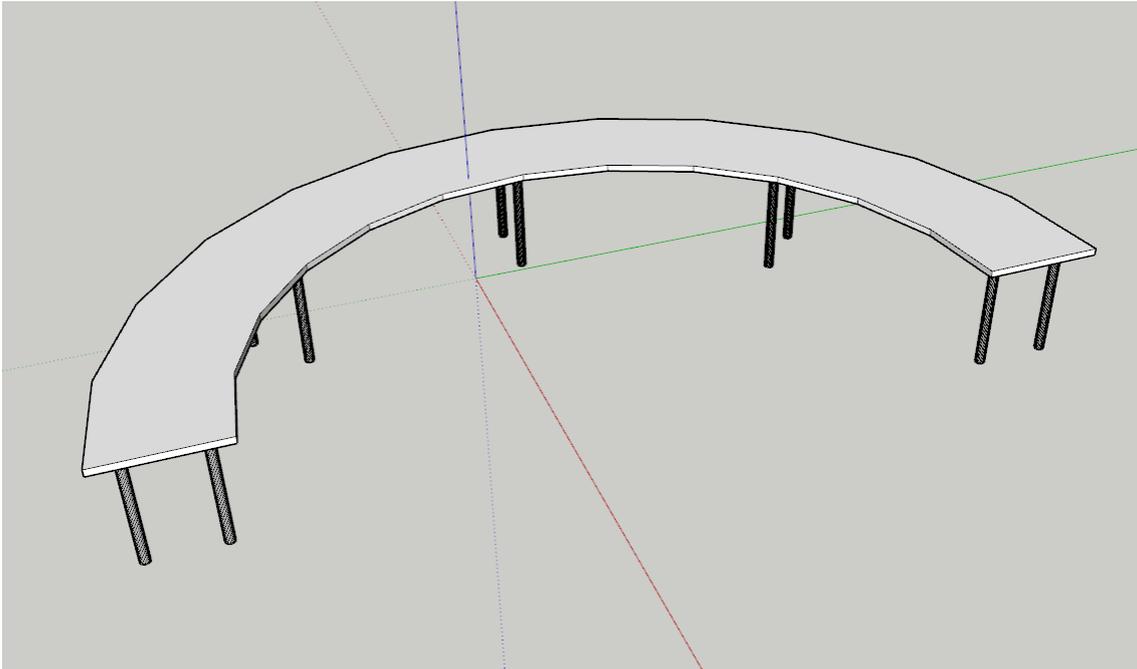
*Ilustración 12 : Modelo 3D del escenario, visto desde arriba.*



*Ilustración 13: Modelo 3D del escenario, visto en perspectiva.*

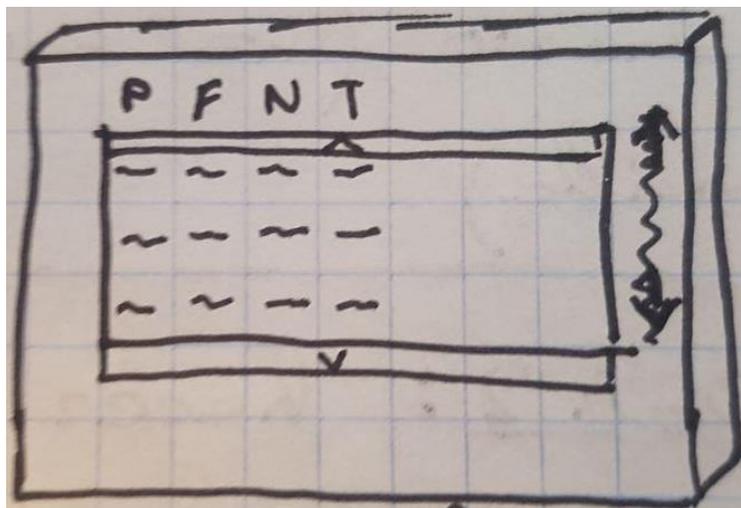
También fue necesario el diseño y modelado 3D, de los siguientes objetos:

- La mesa de la habitación 1, donde se desarrolla el juego, que podemos observar en la **Ilustración 14**.

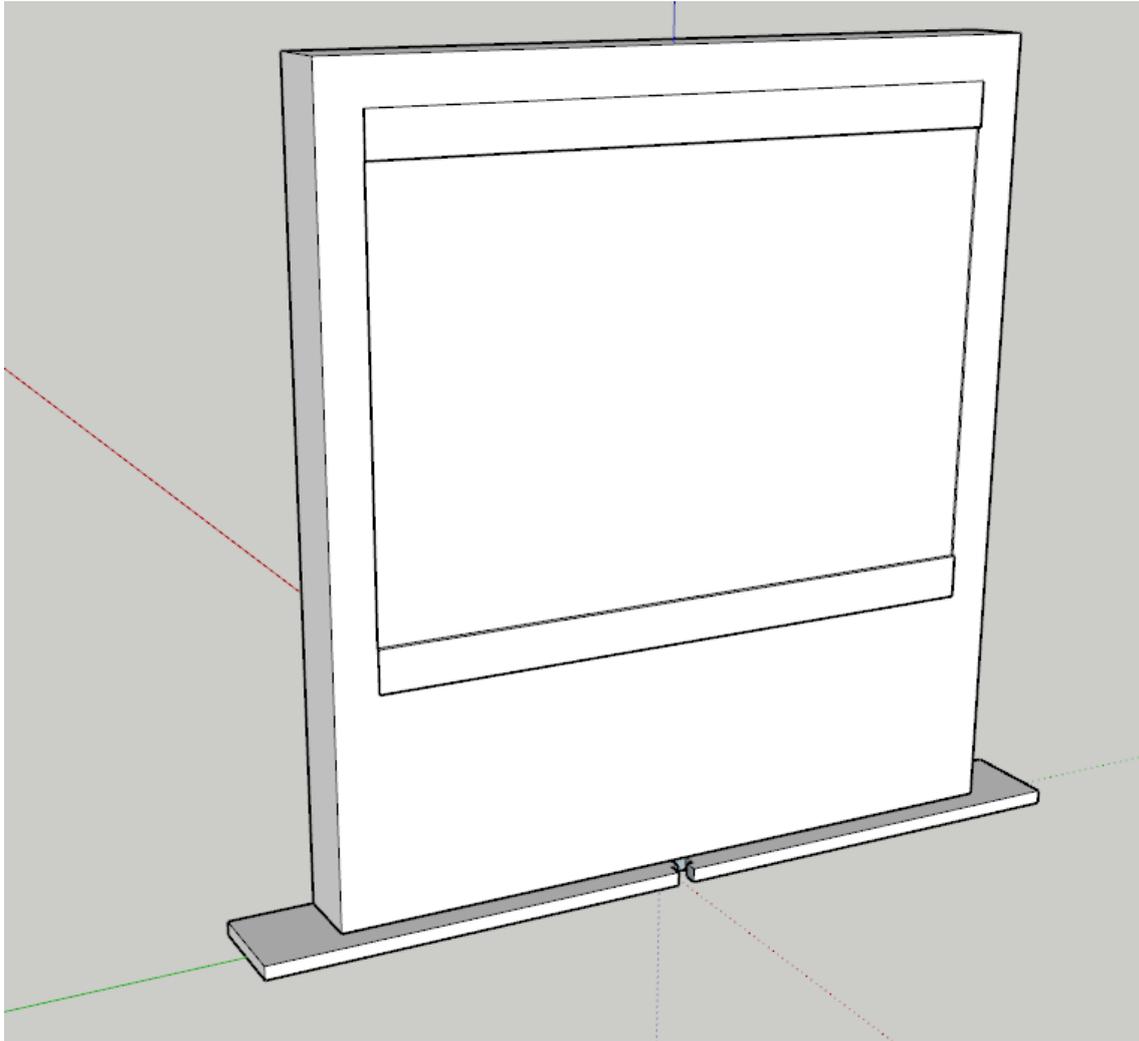


*Ilustración 14: Modelo 3D de la mesa de juego.*

- La tabla de puntuaciones, que fue primero dibujada a mano, **Ilustración 15**, y luego se modeló en 3D como se ve en la **Ilustración 16**.

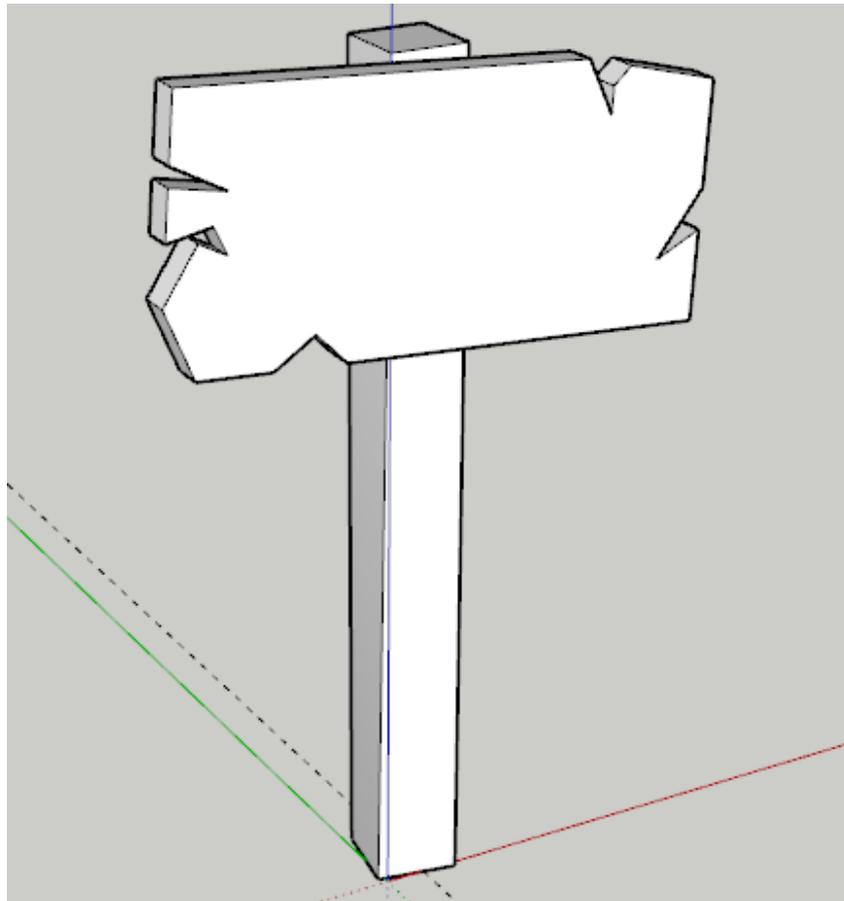


*Ilustración 15: Boceto a mano de la tabla de puntuaciones.*



*Ilustración 16: Modelo 3D de la tabla de puntuaciones.*

- Las señales donde se colocan los mensajes del tutorial, **Ilustración 17**.



*Ilustración 17: Modelo 3D señal de tutoriales.*

En todos estos modelos 3D se les añadió color y animaciones una vez dentro de Unity, utilizando la herramienta correspondiente para cada una de las dos tareas.

### 3.5 Modelo de negocio

El modelo de negocio, que seguiría nuestro juego, sería el que mantienen actualmente muchos de los videojuegos independientes. Esto quiere decir, que pondríamos el juego a la venta antes de que estuviera terminado totalmente, utilizando así a los jugadores como fuente de información para seguir mejorando y a la vez manteniendo el desarrollo con los ingresos obtenidos. Por otro lado, intentaríamos que alguna web o “youtuber<sup>6</sup>” hiciera publicidad, para aumentar la viralidad de nuestro producto.

Para tener una idea global de nuestro producto y de las cosas que ofrecemos con él, nos apoyaremos en el Lean Canvas adaptado a videojuegos de la **Tabla 7**, la versión estándar suele ser la mostrada en la.

<b>Plataforma</b>  -Oculus Rift -GearVr -Windows	<b>Mecánicas</b>  - Cada acierto suma 10 puntos. - Se aumenta el multiplicador de puntuación en 1 unidad por cada acierto. - El multiplicador se reinicia por cada fallo a 1.	<b>Dinámica</b>  - Competitividad. - Progresión.	<b>Estética</b>  - Sencillo. - Competitivo. - Relajante.	<b>Jugadores</b>  - Personas que poseen dispositivos de realidad virtual. - Personas que les guste competir contra si mismos y mejorar.
	<b>Componentes</b>  - Puntuación. - Niveles de dificultad. - Tiempo límite. - Sonidos. - Animaciones. - Tabla de puntuaciones.		<b>Comportamientos</b>  - Responder encuestas. - Leer emails. - Recomendaciones.	
<b>Costes</b>  - Gastos en el desarrollo del videojuego. - Publicidad. - Publicarlo en la Oculus Store.		<b>Ingresos</b>  - Pago por adquisición.		

Tabla 7: Lean Canvas

<sup>6</sup> Persona que se dedica profesionalmente a realizar videos y subirlos a la plataforma de YouTube.

### 3.6 Normativa y legislación

#### 3.6.1 Reglamento que afecta al TFG

El Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales, y su actualización en el Real Decreto 861/2010, de 2 de julio, indica que todas las enseñanzas oficiales concluirán con la elaboración y defensa de un Trabajo Fin de Título (en adelante, TFT) que ha de formar parte del plan de estudios, señala además, que el TFT deberá realizarse en la fase final del plan de estudios y estar orientado a la evaluación de las competencias asociadas al título.

- **Artículo 3.- Definición**

A los efectos de este Reglamento, con la denominación “Trabajo Fin de Título”, se designa a una asignatura que consiste en el desarrollo de un trabajo relativo a las competencias propias de la titulación, realizado de forma autónoma e individualizada por un estudiante universitario, bajo la orientación de un tutor académico que actuará como dinamizador y facilitador del proceso de aprendizaje.

- **Artículo 4.- Finalidad**

El TFT tiene por objetivo elaborar un trabajo en el que el estudiante universitario desarrolle las competencias y los conocimientos adquiridos, teóricos y prácticos como culminación de sus estudios y como preparación para el desempeño futuro de actividades profesionales en el ámbito correspondiente a la titulación obtenida.

#### 3.6.2 Licencias del software

En este proyecto, se hacen uso principalmente de seis elementos software de terceros:

- Unity Personal
- SketchUp
- Visual Studio Community
- ReSharper
- Android Studio
- Oculus Utilities for Unity 5

Cada uno de ellos tiene sus términos legales y sus diferentes licencias del software, las cuales pasaremos a describir a continuación:

- **Unity Personal**

La versión de Unity Personal, permite desarrollar con fines comerciales, siempre que se cumpla el siguiente requisito: es necesario que la compañía que la esté utilizando, no supere los 100.000€ en ingresos brutos anuales o haya recaudado fondos superando esa misma cifra. En este caso, se deberá pasar a otra versión de Unity de pago según la cantidad de ingresos generados por la empresa.

- **SketchUp**

Para poder hacer uso comercial de cualquier material generado por esta aplicación es necesario adquirir una versión Pro de este mismo.

- **Visual Studio Community**

La versión de Visual Studio Community tiene dos tipos de restricciones para su uso, aun las que nos concierne es la primera:

- **Para usuarios individuales:**

Cualquier desarrollador individual puede usar Visual Studio Community para crear sus propias aplicaciones gratuitas o de pago.

- **Para organizaciones**

Un número ilimitado de usuarios dentro de una organización puede usar Visual Studio Community para los siguientes escenarios: en un entorno aprendizaje en clase, para investigación académica o para contribuir en proyectos de código abierto.

- **ReSharper**

La versión de ReSharper de estudiante, deja terminantemente prohibida la comercialización de cualquier producto desarrollador con esta aplicación.

- **Android Studio**

- **Oculus Utilities for Unity 5**

Tanto Android Studio, como Oculus Utilities for Unity 5, no ponen ningún tipo de impedimento a la hora de comercialización de los productos realizados con ambos softwares.

## 4 Desarrollo

### 4.1 Alcance de la implementación

A continuación, se pasará a describir las características que fueron implementadas durante el desarrollo del proyecto:

- El jugador puede continuar su anterior partida desde el menú de carga accionando el botón “Play Game”.
- El jugador puede crear una nueva partida y empezar de cero desde el menú de carga accionando el botón de “New Game”, después se mostrará un dialogo para confirmar si se desea borrar la anterior partida.
- El jugador puede moverse por el mundo en los transportadores, **Ilustración 18**, enfocando la mirada a ellos y dándole al botón de acción.



*Ilustración 18: Transportador enfocado con la mirada.*

- El jugador, si dispone de un mando, puede usar este para moverse, activándolo desde el menú. Solo en Oculus Rift, no en GearVr.
- El jugador puede iniciar un nivel, entrando en alguna de las habitaciones y buscando el panel de control de ese nivel y accionando el botón “Start”, **Ilustración 19**.
- El jugador puede desbloquear diferentes niveles según avanza y completa cada nivel.
- El jugador puede elegir la dificultad del nivel desde el panel de control, **Ilustración 19**.



*Ilustración 19: Panel de control, habitación número 1*

- El jugador puede abortar un nivel, abriendo el panel de control y accionando el botón "Stop", **Ilustración 19**.
- El jugador puede observar sus mejores puntuaciones, abriendo el panel de puntuaciones, accionando el dispositivo de la **Ilustración 20**, que abrirá el panel de la **Ilustración 21**.



*Ilustración 20: Consola para abrir el menú de puntuaciones.*



*Ilustración 21: Panel de máximas puntuaciones.*

- El jugador tendrá un tutorial que le explicará cómo funciona el juego, los paneles de los tutoriales tienen la apariencia de la **Ilustración 22**.



*Ilustración 22: Panel de tutorial.*

- El jugador puede descartar cada tutorial, enfocando la mirada hacia él y dándole al botón de acción.
- El jugador puede abrir un menú de opciones clicando en el botón de atrás del mando, el menú tiene la apariencia de la **Ilustración 23**.



*Ilustración 23: Menú interno del juego.*

- El jugador puede volver al gestor de partidas, desde el menú interno de la **Ilustración 23** y accionando el botón “Exit”.
- El jugador puede eliminar todos los tutoriales, accionando el botón “Remove Tutorials” del menú de la **Ilustración 23**.
- El jugador puede activar o desactivar el sonido del juego desde el menú de la **Ilustración 23** y accionando el botón “Mute Sound”/”Unmute Sound”.
- El sistema guardará el estado del juego cada vez que se termine un nivel, y se almacenará en el archivo de datos.
- El sistema actualizará la tabla de marcadores cada vez que termine un nivel.
- El sistema detecta el tipo de dispositivo:
  - Si es un sistema de realidad virtual, cede el control de la cámara a las gafas.
  - Si es un sistema sin realidad virtual, cede el control de la cámara al ratón.
- El sistema reproduce un sonido cuando se interactúa con un objeto.
- El sistema reproduce un sonido cuando se pasa el cursor por encima de un objeto con el que se puede interaccionar.
- El sistema reproduce un sonido cuando se termina satisfactoriamente una partida.
- El sistema reproduce un sonido cuando no se termina de forma satisfactoria una partida.
- El sistema bloquea la puerta de cada nivel hasta que se termine el nivel donde se encuentra.

#### 4.2 Diseño arquitectónico

Unity se basa en un sistema por componentes, como ya se ha comentado anteriormente. Por lo tanto, el sistema está formado por una serie de componentes principales. Uno de los componentes principales es el jugador en sí, el cual está implementado de forma que se adapte tanto a dispositivos VR, como a los que no, a parte permite la utilización o no de un mando. Todo esto lo gestiona este componente y se puede exportar a cualquier otro proyecto. Además, si es necesario realizar cualquier acción sobre el jugador, se deberá pasar por este componente.

Por otro lado, cada menú es también un componente independiente, el cual se encarga de gestionar cada uno de sus botones, este componente puede interaccionar con otros componentes si alguna acción así lo requiere. Por ejemplo, el botón de

activar/desactivar el mando, debe interactuar con el componente del jugador para poder realizar dicha acción.

También nos encontramos con el componente encargado de controlar el juego en sí. Este componente conoce los diferentes niveles implementados en el juego, los cuales deben implementar la interfaz *RoomGame* y asociarle la puerta de dicho nivel. Cuando se inicia un nivel, se bloquea la puerta de ese nivel para que el jugador no pueda salir mientras se desarrolla la partida, en cualquier momento se puede abortar la partida para poder salir del nivel. Además, al completar el nivel, se encarga de guardar el estado del juego en un archivo externo.

Por lo tanto, estamos ante una arquitectura por componentes los cuales pueden coexistir por sí solos o interactuar entre ellos si así se necesita, estos componentes se pueden exportar en diferentes proyectos con gran facilidad y reutilizarlos, para así ahorrar tiempo de tener que volver a implementarlo.

#### 4.3 Modelo de la base de datos

Este proyecto no posee una base de datos. La forma de mantener el anterior estado del juego es un archivo binario con la estructura de datos de la **Ilustración 24**.

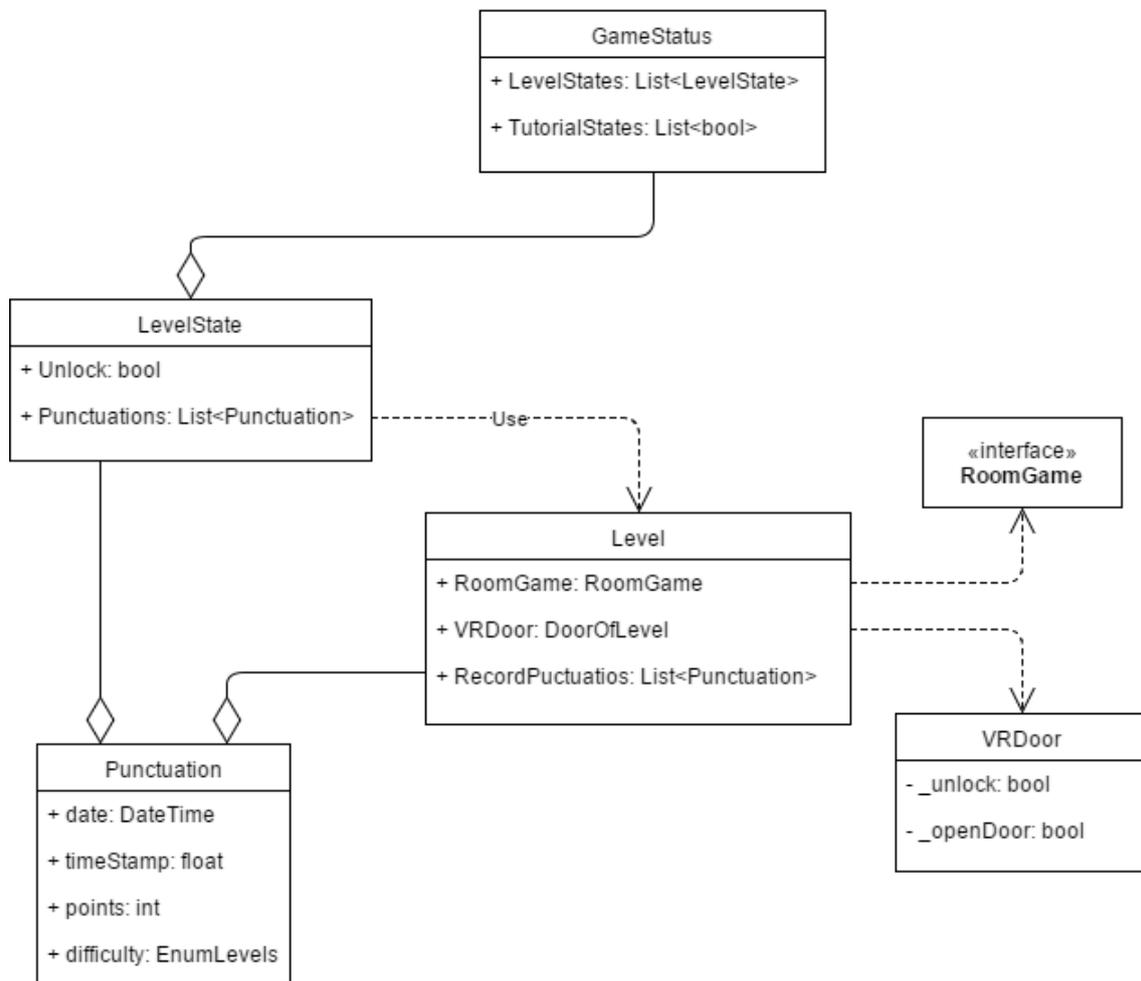


Ilustración 24: Diagrama de clases de la estructura de datos.

Se descartó la idea de usar una base de datos, por las razones que se exponen a continuación:

- No iba a depender de una conexión a internet o un servidor local. Muchas bases de datos como MySQL necesitan de un servidor para poder ser utilizadas, por lo tanto, sería necesario conectarse remotamente a ella o ejecutar un servidor local.
- Era necesario que la información almacenada no fuera fácil de modificar. Al descartar la idea de usar un servidor, el archivo local debería ser difícil de modificar. Por lo tanto, tampoco era una opción SQLite, pues simplemente utilizando el programa de SQLite se puede ver y modificar de forma sencilla.

En un principio se optó por guardarlo en un archivo XML, el problema es que facilitaba a cualquier usuario a conocer la forma en que se guardan los datos, como se ve en la **Ilustración 25**, y poder hacer trampas en el juego, añadiéndose mejores puntuaciones o desbloqueando el resto de niveles. Al guárdalo en un archivo binario, si no se conoce exactamente la estructura de datos es muy complicado decodificar el archivo y poder escribir en él.

```

    <?xml version="1.0" encoding="UTF-8" ?>
    - <Data>
      - <Tabla2>
        <Nombre>Juan</Nombre>
        <Apellido>Casas</Apellido>
        <ID>123.4</ID>
      </Tabla2>
      - <Tabla2>
        <Nombre>Pedro</Nombre>
        <Apellido>Diaz</Apellido>
        <ID>789.56</ID>
      </Tabla2>
    </Data>
  
```

Diagram annotations:

- refElement**: points to the opening tag of the first `<Tabla2>`.
- valorElementHijo**: points to the text content `Juan` inside the `<Nombre>` tag of the first `<Tabla2>`.
- nomElementHijo**: points to the opening tag of the `<Nombre>` tag of the first `<Tabla2>`.

*Ilustración 25: Estructura de un archivo XML*

En la **Ilustración 26** sería como se ve el archivo, donde se guarda la partida. Si lo abrimos con un lector de texto, (en nuestro caso el Notepad++) el cual acepta diferentes tipos de archivos de texto. El programa intenta transformar la información guardada, en caracteres ASCII y como resultado de ello se ven símbolos y letras sin sentido aparente.



de programación. Para modelar los objetos en 3D se ha optado por la herramienta SketchUp, la cual nos permite exportar a Collada (.dae) y luego importarlo desde Unity.

#### 4.4.1 C#

Es un lenguaje de programación que apareció en el año 2000 y actualmente va por la versión 6.0. Es un lenguaje de programación orientado a objetos, desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

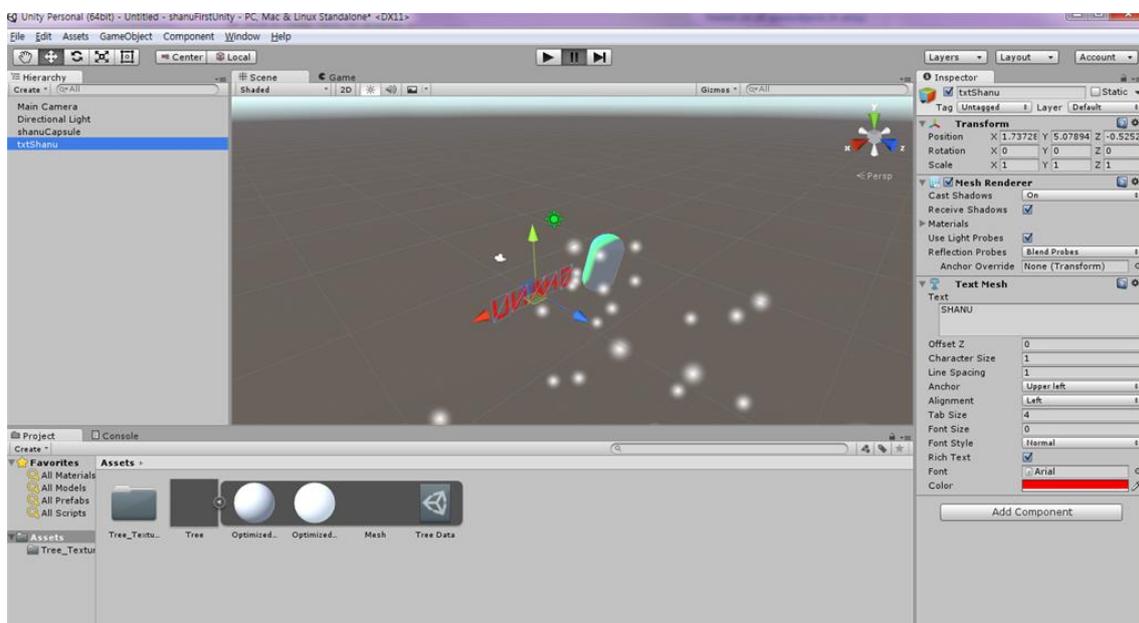
#### 4.4.2 Unity Personal

Es un motor de videojuego multiplataforma creado por Unity Technologies, que fue lanzado el 30 de mayo de 2005 y actualmente se encuentra en la versión 5.6 estable. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. A partir de su versión 5.4.0 ya no soporta el desarrollo de contenido para navegador a través de su plugin web, en su lugar se utiliza WebGL. Unity tiene dos versiones: Unity Professional (pro) y Unity Personal.

La primera versión de Unity se lanzó en la Conferencia Mundial de Desarrolladores de Apple en 2005. Fue construido exclusivamente para funcionar y generar proyectos en los equipos de la plataforma Mac y obtuvo el éxito suficiente como para continuar con el desarrollo del motor y herramientas. Unity 3 fue lanzado en septiembre de 2010 y se centró en empezar a introducir más herramientas que los estudios de alta gama por lo general tienen a su disposición, con el fin de captar el interés de los desarrolladores más grandes, mientras que proporciona herramientas

para equipos independientes y más pequeñas que normalmente serían difíciles de conseguir en un paquete asequible.

En la **Ilustración 27** se muestra como es la interfaz de usuario de la aplicación. Como se puede observar en la parte central, se encuentra la zona donde se colocan los objetos en el escenario, a la derecha se pueden observar las características de cada objeto, en la parte izquierda se encuentra la jerarquía de los objetos dentro de la escena y en la parte inferior el navegador de archivos y la consola. Aunque es totalmente personalizable al gusto de cada desarrollador.



*Ilustración 27: Interfaz gráfica predeterminada de Unity.*

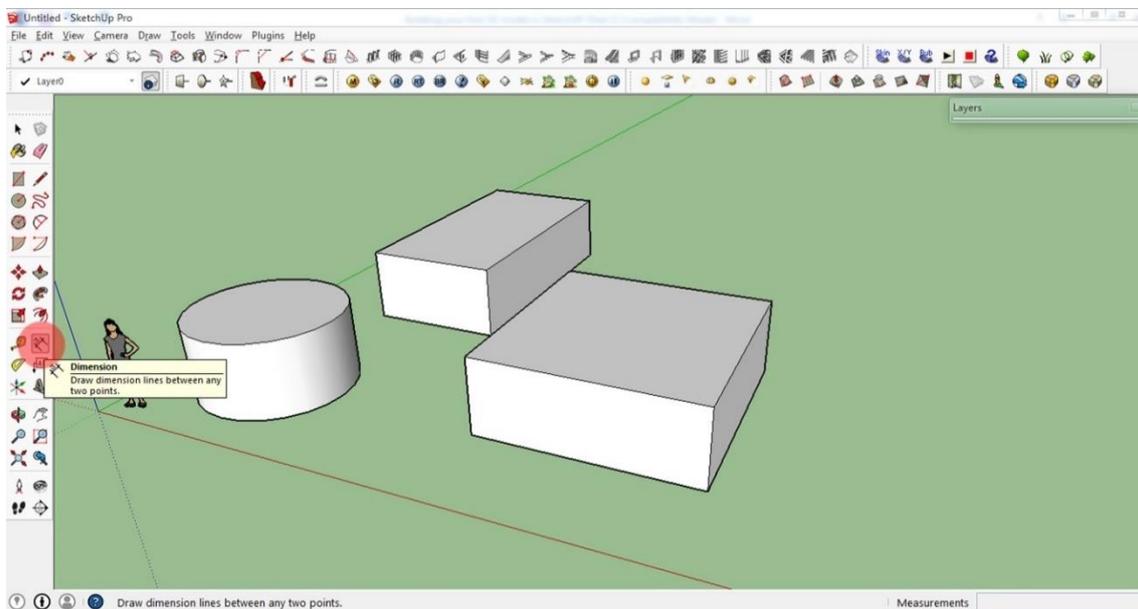
#### 4.4.3 SketchUp

es un programa de diseño gráfico y modelado en tres dimensiones (3D) basado en caras. Su fecha de lanzamiento es agosto de 2000 y actualmente se encuentra en la versión 16.1. Está pensado para entornos de arquitectura, ingeniería civil, diseño industrial, diseño escénico, GIS, videojuegos o películas. Es un programa desarrollado por @Last Software, empresa adquirida por Google en 2006 y finalmente vendida a Trimble en 2012.

Su principal característica es poder realizar diseños en 3D de forma sencilla. El programa incluye entre sus recursos un tutorial en vídeo para aprender paso a paso cómo se puede diseñar y modelar el propio ambiente. Permite conceptualizar y

modelar imágenes en 3D de edificios, coches, personas y cualquier objeto o artículo que imagine el diseñador o dibujante, además de que el programa incluye una galería de objetos, texturas e imágenes listas para descargar.

En la **Ilustración 28** se muestra como es la interfaz de usuario de la aplicación. Al ser un programa de modelado 3D la mayoría de la interfaz es una ventana donde utilizar las diferentes herramientas, que estas distribuidas por los laterales de esa misma ventana. Las barras de herramientas se pueden colocar en la posición que más cómodo sea para el usuario.



*Ilustración 28: Interfaz de usuario del SketchUp.*

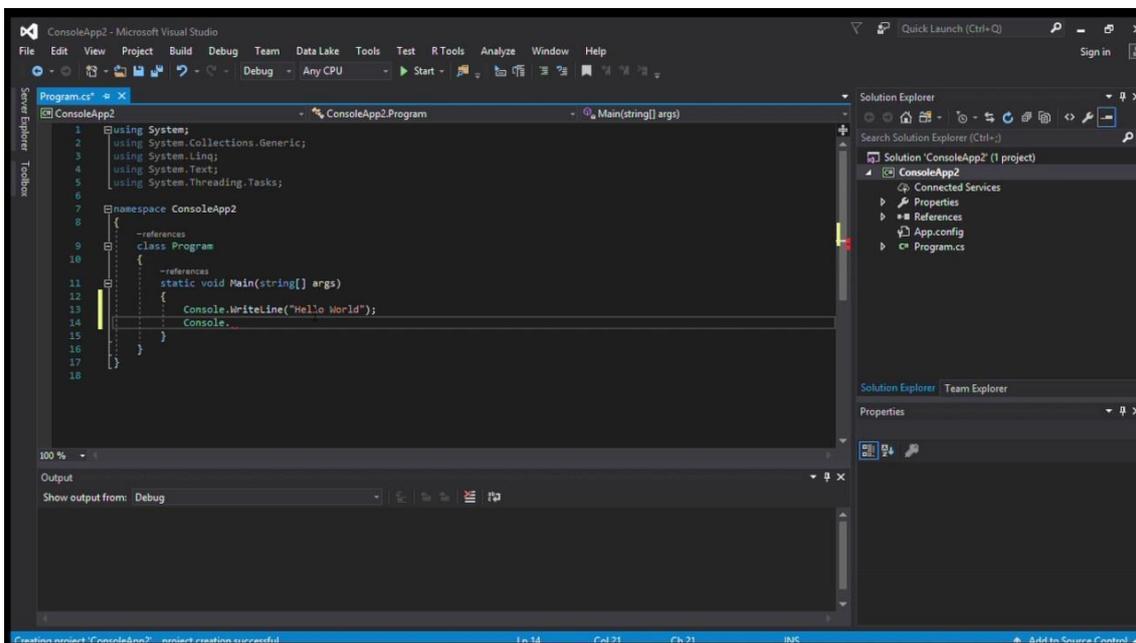
#### 4.4.4 Visual Studio

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Su fecha de lanzamiento fue 1 de mayo de 1997 y actualmente se encuentra por la versión 2017. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc.

A partir de la versión 2005, Microsoft ofrece gratuitamente las Ediciones Express, que son versiones básicas separadas por lenguajes de programación o plataforma enfocadas a estudiantes y programación amateur. Estas ediciones son

iguales que el entorno de desarrollo comercial, pero sin características avanzadas de integración.

En la **Ilustración 29** se puede observar la interfaz de usuario del Visual Studio. Como en la mayoría de los IDEs<sup>7</sup> en el espacio central se dispone la ventana donde escribir las líneas de código, justo debajo se encuentra la consola y en la parte derecha de la aplicación el navegador de archivos. Como en la mayoría de los IDEs este se puede personalizar tanto en color, como en aspecto, pudiendo ajustar el tamaño de cada zona, añadir diferentes paneles y cambiarlos de lugar.



*Ilustración 29: Interfaz de usuario de Visual Studio predeterminada.*

#### 4.5 Pruebas

Durante el desarrollo de la aplicación, se realizaron numerosas pruebas, para comprobar que la calidad era siempre la esperada. La mayoría de las pruebas se basaron principalmente en ejecutar la aplicación y simular diferentes escenarios posibles.

---

<sup>7</sup> Del inglés Integrated Development Environment, entorno de desarrollo integrado, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Para ellos fue necesario que usuarios reales probaran la aplicación y detectaran errores durante su ejecución, con esto también se conseguía un *feedback*<sup>8</sup> importante, para poder mejorar más la aplicación.

Al ser un entorno 3D realizar pruebas programáticas era más complicado que encontrar potenciales usuarios que pudieran probar nuestra aplicación, reportarnos los errores encontrados y proporcionarnos información muy valiosa sobre cómo mejorar la aplicación o simplemente comentarnos que aspectos les gustaban más.

Los usuarios escogidos para este tipo de pruebas fueron tanto compañeros de clase como compañeros de la empresa.

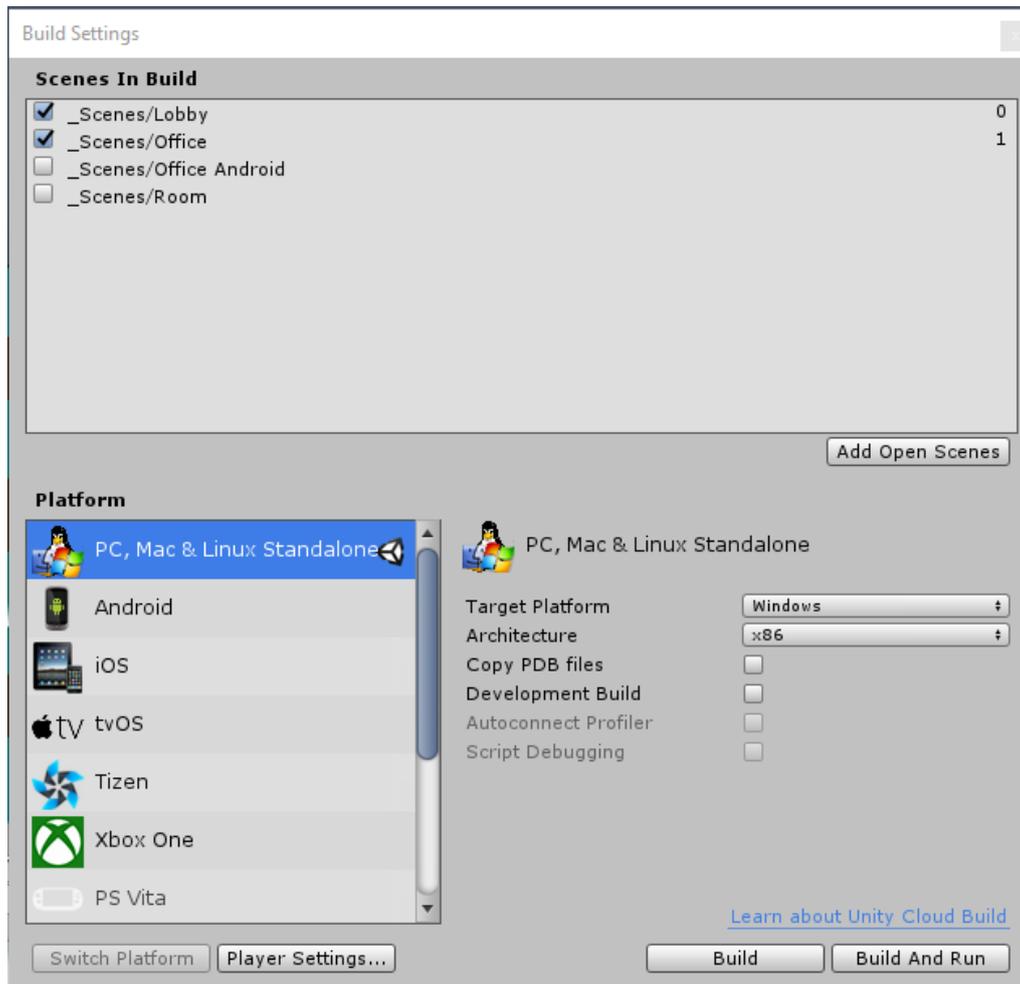
Ambos colectivos reportaron diferentes fallos, estéticos, técnicos o simplemente aportaron su opinión y dieron ideas para mejorar el proyecto.

#### 4.6 Despliegue

Gracias a la herramienta Unity, el despliegue en diferentes plataformas se facilita considerablemente, esta misma dispone de un panel (**Ilustración 30**), que permite seleccionar para que plataforma deseas desplegar la aplicación y en un solo clic se pone en marcha.

---

<sup>8</sup> Retroalimentación, conjunto de reacciones o respuestas que manifiesta un receptor respecto a la actuación del emisor, lo que es tenido en cuenta por este para cambiar o modificar su mensaje.



*Ilustración 30: Panel de despliegue de Unity.*

#### 4.6.1 Despliegue para equipos de sobremesa.

En las siguientes líneas se pasará a describir los diferentes pasos que se deben seguir si queremos desplegar nuestra aplicación:

1. Primero, una vez dentro de la herramienta Unity iremos a la barra superior e iremos a **File>Build Settings...**, esto nos llevará al panel de la **Ilustración 30**, que ya vimos anteriormente.
2. A continuación, deberemos seleccionar la plataforma **PC, Mac & Standalone**.
3. Si venimos de otra plataforma, deberemos dar clic en **Switch Platform**.
4. Por último, le daremos al botón de **Build** y nos saldrá un dialogo de donde deseamos que se guarde nuestra aplicación ya desplegada en el lugar elegido.

Una vez concluido estos pasos, solo será necesario dirigirnos al lugar donde se realizó el despliegue y ejecutar la aplicación resultante del proceso.

#### 4.6.2 Despliegue para equipos móviles

Si se quiere desplegar en dispositivo móvil, los pasos a seguir para desplegarlo son parecidos, pues se debe seleccionar la plataforma **Android**. Además, se debe realizar una serie de configuraciones, debido a que tanto Oculus como Android no permiten ejecutar cualquier tipo de aplicación para evitar así posibles problemas de seguridad. Por lo tanto, ahora se pasará a explicar que pasos se deben llevar a cabo si se quiere hacer un despliegue correcto en Android:

1. Se debe bajar el SDK de Android, el Android Studio de la web <https://developer.android.com>
2. Especificar dentro de **Unity > Preferences > External Tools** y se abre el dialogo de la **Ilustración 31**.
3. En el apartado SDK, del dialogo de la **Ilustración 31**, hay que especificar la ruta donde se instaló el SDK de Android, en nuestro caso la ruta fue **Appdata/Local/Android/**.

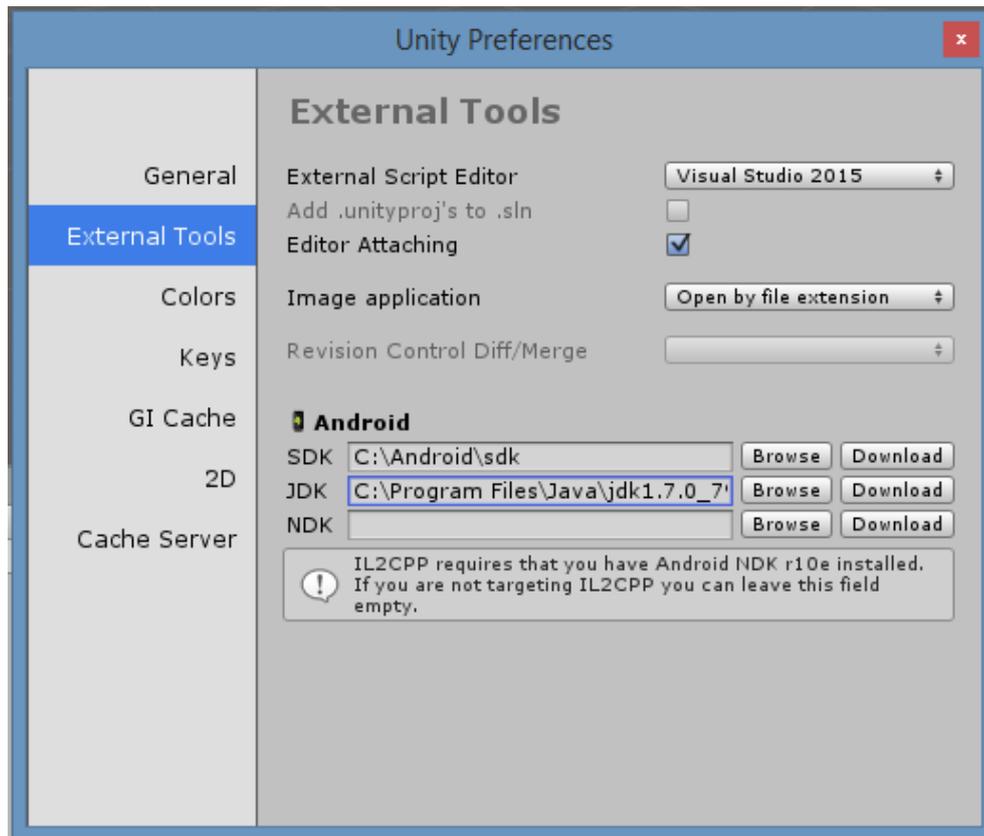


Ilustración 31: Dialogo de External Tools

Una vez terminado estos pasos se puede realizar un despliegue de la aplicación, pero a la hora de intentar ejecutarla en Android da un fallo, debido a que la aplicación no se encuentra firmada por Oculus. Para eso debemos seguir los siguientes pasos:

1. Debemos acceder al siguiente link <https://dashboard.oculus.com/tools/osig-generator/>, ahí deberemos iniciar sesión y se nos mostrará algo como en la **Ilustración 32**.



Ilustración 32: Página web de Oculus para firmar las aplicaciones.

2. Tenemos que obtener el ID de nuestro dispositivo Android. Para poder introducirlo en la página, deberemos ejecutar un terminal del sistema operativo y ejecutar el comando siguiente, con el dispositivo conectado a la máquina:

```
adb devices
```

Esto debería mostrar algo parecido a lo siguiente:

```
List of devices attached  
ce0551e7 device
```

Donde “ce0551e7” sería el id del dispositivo.

3. Nos volvemos a dirigir a la página del primer paso e introducimos el ID obtenido desde la consola.
4. La página descargará un fichero, que es la firma. Este se debe introducir dentro del proyecto en el directorio **Assets/plugins/Android/Assets/**, en caso de no existir, se deberá crear el directorio y mover el archivo descargado a dicho lugar.

Ahora ya se puede desplegar y ejecutar nuestra aplicación en un dispositivo móvil.

## 5 Conclusiones

Sobre el año 2010 empezó el fenómeno de las televisiones 3D, de aquella todo el mundo pensaba que eso sería el futuro y que no podría ser mejor. Ahora en 2017 se repite la historia, todo el mundo cree que esto será el futuro, pero yo creo que esto no acaba aquí, que esto es solo un pequeño paso antes de algo mucho más grande, que desde mi opinión creo que serán los hologramas.

Llego a esta conclusión porque a mi parecer por mucho que solucionaran el tema de los cables del dispositivo, lo hicieran más ligero, mejoraran la calidad..., seguiría siendo un sistema muy mejorable, pues solamente puedes ver una persona las cosas y para interactuar dependes de mandos que recojan tus movimientos. La gente lo que desea es lo más sencillo, que le dé a un botón y funcione. Puede que lleguen a coexistir las dos tecnologías al mismo tiempo, pues tienen utilidades diferentes, por ejemplo, aun se siguen emitiendo películas en el cine en 3D y existen las gafas de realidad virtual.

Por nuestra parte hemos intentado aportar algo nuevo a esta tecnología y comprender un poco más como funciona por dentro, como se interactúa y que posibilidades tiene.

La verdad que es una maravilla. No tiene nada que ver con lo que estamos acostumbrados a utilizar, pues con esto dejas muy lejos los actuales sistemas de interacción humano-máquina, que no nos permiten relacionarnos con la máquina de una forma natural.

Todo esto nos llevó a plantearnos como sería desarrollar una aplicación para este tipo de tecnología, que cosas serían necesarias, como de complicado sería adaptarse a este nuevo sistema de interacción. También decir que, durante los estudios, nunca se había recibido ningún tipo de enseñanza sobre cómo utilizar motores gráficos. Por lo tanto, fue todo un reto aprender a utilizar esta nueva herramienta de desarrollo, la cual no se parecía para nada a ninguna utilizada durante nuestra enseñanza.

Han sido muchas horas leyendo información, haciendo tutoriales de Unity para aprender a utilizar lo más básico al menos, realizando pruebas, pensando como diseñar el juego, el escenario, las animaciones, los sonidos, etc. Son muchas cosas que tener en cuenta cuando se hace un videojuego. No basta tener una buena idea, si no se construye un buen escenario, con una armonía y con una buena ambientación, pues

se necesita que el usuario se sienta lo más cómodo posible dentro de este nuevo mundo.

También decir que, durante el desarrollo surgieron múltiples ideas que hubieran sido fantásticas introducirlas, pero por falta de tiempo y conocimientos avanzados sobre estas tecnologías se descartaron, pero no quiere decir que, en un futuro se siga desarrollando y se implementen estas ideas u otras más nuevas y mejores.

Esperemos que esta tecnología siga avanzando como hasta ahora y que nos siga sorprendiendo. Pues la curiosidad siempre es insaciable y siempre se quiere un poco más.

### 5.1 Líneas futuras

Durante el transcurso de este proyecto, solo hemos creado los cimientos de lo que se espera sea una gran estructura.

En el futuro, nos gustaría añadir diferentes modos de juego, y que cada modo de juego vaya añadiendo diferentes formas de interaccionar con los dispositivos de realidad virtual, de una forma progresiva, para así enseñarle al jugador que cosas se pueden hacer con esta nueva tecnología y que a la vez sea entretenido. Por ejemplo, nos hubiera gustado implementar un modo de juego en la segunda habitación, en el cual se tendría que reconstruir una escena previamente mostrada. Para ello se debería poder mover los objetos de un lugar a otro de la habitación (ya implementado, pero falta mejorarlo).

Cualquier tipo de aplicación debe ser atractiva, si queremos que tenga una gran captación entre el público y aún más siendo un videojuego. Por lo tanto, nos gustaría poder mejorar el apartado gráfico, es decir, las luces, sombras, animaciones, efectos visuales y rediseñar algunos objetos del escenario. Para que así, el usuario se sienta más cómodo y atraído por el escenario, pero siempre manteniendo ese aspecto de dibujos sencillos, pues es la esencia del juego.

## 6 Anexos

### 6.1 Anexo I: Manual de usuario

Para facilitar la instalación y utilización del juego se ha realizado un manual de usuario explicando de forma resumida y sencilla que se debe hacer para poder ejecutar la aplicación en cada una de las plataformas disponibles, cómo funciona el menú principal y los controles básicos. Todo esto se puede observar en las siguientes ilustraciones:

- **Ilustración 33: Portada del manual de usuario.**
- **Ilustración 34: Primera página del manual, con la instalación y menú principal.**
- **Ilustración 35: Segunda página del manual, con el menú de partida.**
- **Ilustración 36: Tercera página del manual, con los controles.**



*Ilustración 33: Portada del manual de usuario.*

## Primeros pasos

---

### Instalación

#### Dispositivos móviles

Copiar el APK resultante del despliegue en el móvil compatible con la tecnología de Oculus. Buscar con el explorador de archivos e instalar. Se deberá habilitar la opción de Fuentes desconocidas, para permitir que se instale la aplicación.

#### Dispositivos de escritorio

Copiar la carpeta resultante del despliegue en el directorio deseado.

## Menú principal

---

Ejecutar la aplicación instalada en caso del dispositivo móvil buscar la aplicación "Memorize-Vr", en el cajón de aplicaciones, y en caso del dispositivo de escritorio, acceder a la carpeta y ejecutar el "Memorize-Vr.exe". Una vez terminada la carga del juego aparecerá una imagen como la siguiente, con las siguientes opciones:

- New Game: borra la anterior partida y comienza una nueva.
- Play Game: continua con la última partida existente.
- Exit Game: cierra el juego y vuelve al escritorio.



1

*Ilustración 34: Primera página del manual, con la instalación y menú principal.*

## Menú de partida

Una vez dentro del juego se puede abrir un menú con el botón dedicado para ellos y el cual tiene el aspecto de la imagen siguiente, con las siguientes opciones:



- Disable Controller / Enable Controller: activa o desactiva la opción de moverse con el mando por el escanario.
- Remove Tutorials: elimina todos los tutoriales de la partida.
- Mute Sound / Unmute Sound: activa o desactiva el sonido del juego
- Exit: guarda la partida y vuelve al menú de carga.

2

*Ilustración 35: Segunda página del manual, con el menú de partida.*

## Controles

---

### Controles con mando

Joystick izquierdo: movimiento del personaje.

A: interactuar.

B: abre el menú.



### Controles con GearVR



Home: vuelve al menú de Oculus.

Return: abre el menú.

Touchpad: un toque para interactuar.

3

*Ilustración 36: Tercera página del manual, con los controles.*

## 6.2 Anexo II: Referencias

Para que el lector pueda ampliar sus conocimientos sobre los expuesto en este documento se da paso a los links consultados durante el desarrollo del proyecto, todos estos links se encuentran disponibles a fecha de 10/07/2017. Se han dividido en diferentes categorías para facilitar su organización y poder buscar con mayor facilidad.

### 6.2.1 Consultas terminológicas

- Wikipedia
  - <https://es.wikipedia.org/wiki/Wikipedia:Portada>
- RAE
  - <http://dle.rae.es/?id=DgIqVCc>

### 6.2.2 Recursos: Software

- Visual Studio
  - <https://www.visualstudio.com/es/vs/>
  - [https://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://es.wikipedia.org/wiki/Microsoft_Visual_Studio)
- Unity 5.5
  - <https://unity3d.com/es/unity>
  - [https://es.wikipedia.org/wiki/Unity\\_\(motor\\_de\\_juego\)](https://es.wikipedia.org/wiki/Unity_(motor_de_juego))
- ProjectLibre
  - <http://www.projectlibre.com/product/projectlibre-open-source>
  - <https://es.wikipedia.org/wiki/ProjectLibre>
- SketchUp
  - <http://www.sketchup.com/es/products/sketchup-pro>
  - <https://es.wikipedia.org/wiki/SketchUp>
- Android Studio
  - <https://developer.android.com/studio/index.html>
- C#
  - [https://es.wikipedia.org/wiki/C\\_Sharp](https://es.wikipedia.org/wiki/C_Sharp)

### 6.2.3 Hardware

- Realidad Virtual
  - [https://es.wikipedia.org/wiki/Realidad\\_virtual](https://es.wikipedia.org/wiki/Realidad_virtual)
- Oculus Rift
  - <https://www.oculus.com/rift/>

- [https://es.wikipedia.org/wiki/Oculus\\_Rift](https://es.wikipedia.org/wiki/Oculus_Rift)
- GearVR
  - [https://es.wikipedia.org/wiki/Samsung\\_Gear\\_VR](https://es.wikipedia.org/wiki/Samsung_Gear_VR)
  - <http://www.samsung.com/es/wearables/gear-vr-sm-r324/>

#### 6.2.4 Tutoriales

- Tutoriales de Unity 3D
  - <https://unity3d.com/es/learn/tutorials>

#### 6.2.5 Normativa y legislación

- Unity 3D
  - <https://unity3d.com/es/legal/eula>
- SketchUp
  - <https://www.sketchup.com/es/license/c/sketchup>
- Visual Studio
  - <https://www.visualstudio.com/es/vs/community/>
- ReSharper
  - [https://www.jetbrains.com/student/license\\_educational.html](https://www.jetbrains.com/student/license_educational.html)
- Android Studio
  - <https://developer.android.com/legal.html?hl=es-419>
- Oculus Utilities for Unity 5
  - <https://developer.oculus.com/licenses/>

#### 6.3 Anexo III: Logos

Cuando instalamos una aplicación, tanto en el ordenador, como en el móvil, siempre lo que buscamos es el icono de la aplicación, que suele ser un logo representativo de esta misma, pues nos resulta más sencillo que leernos los nombres de cada una de las aplicaciones instaladas. Por lo tanto, se realizó un diseño, más o menos básico, de un logo y un icono para esta misma. De esta forma se le aporta una mayor formalidad al proyecto y se facilita que el usuario la encuentre sin tener que buscar por todas sus aplicaciones leyendo los nombres.

En la **Ilustración 37** se puede observar el primer diseño del logo. Después, fue sustituido por el de la **Ilustración 38** el cual es el actual logo de la aplicación. También



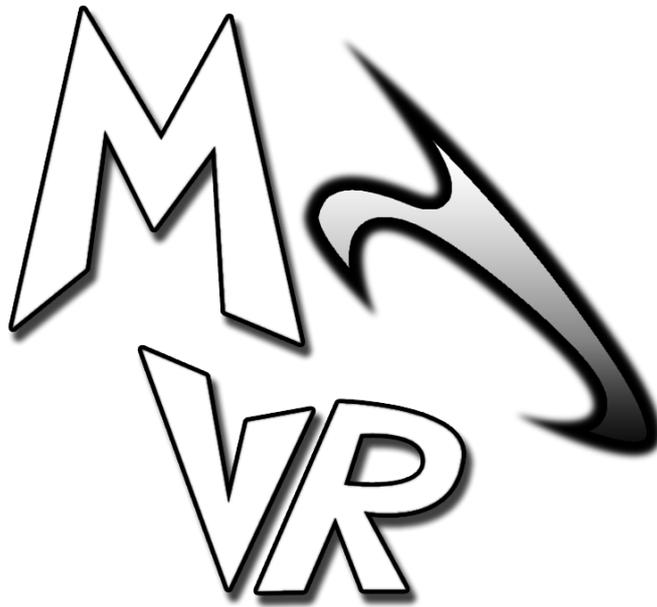
se le añadió un fondo, para darle más aspecto de icono, **Ilustración 39**. Para realizar estos logos se hizo uso de la herramienta GIMP<sup>9</sup>.



*Ilustración 37: Primera versión del logo.*

---

<sup>9</sup> Son las siglas de GNU Image Manipulation Program. Es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Forma parte del proyecto GNU.



*Ilustración 38: Segunda versión del logo.*



*Ilustración 39: Icono de la segunda versión del logo.*